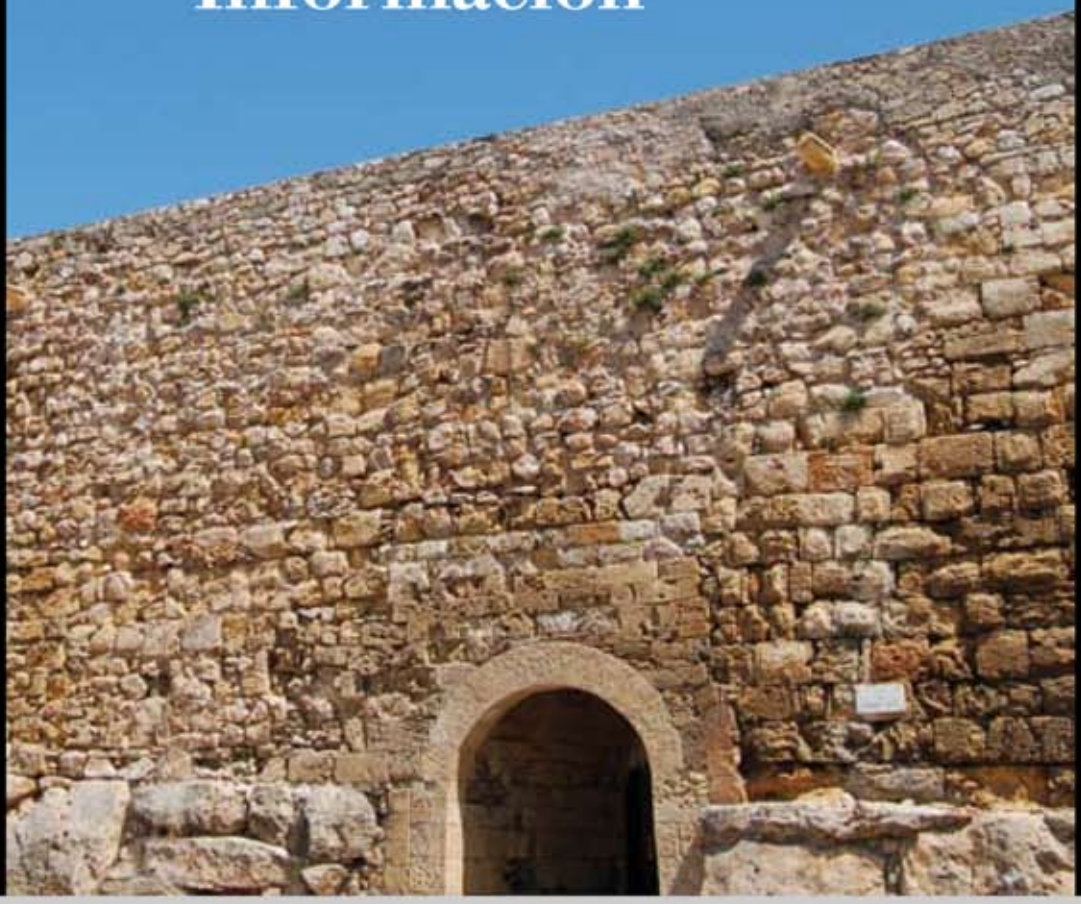
Universitat Rovira i Virgili

RECSI 2010

Tarragona,
7–10 septiembre 2010

Coordinado por:
Josep Domingo Ferrer, Antoni Martínez Ballesté,
Jordi Castellà Roca, Agustí Solanas Gómez

XI Reunión Española sobre Criptología y Seguridad de la Información



[publicacions] urv

RECSI 2010

IX Reunión Española sobre Criptología y Seguridad de la Información

Tarragona 7–10 de septiembre 2010

Coordinado por:
Josep Domingo Ferrer
Antoni Martínez Ballesté
Jordi Castellà Roca
Agustí Solanas Gómez

[publicacions] urv
Tarragona, 2010

Edita:
Publicacions URV

1º edició: juliol 2010

Impressió:
Depòsit Legal:
ISBN: 978-84-693-3304-4

Publicacions de la Universitat Rovira i Virgili:
Av. Catalunya, 35 - 43002 Tarragona
Tel. 977 558 474 - Fax: 977 558 393
www.urv.cat/publicacions
publicacions@urv.cat

Arola Editors: Polígon Francolí, parcel·la 3, nau 5 - 43006 Tarragona
Tel. 977 553 707 - Fax 977 542 721
arola@arolaeditors.com

Cossetània Edicions: C. de la Violeta, 6 - 43800 Valls
Tel. 977 602 591 - Fax 977 614 357
www.cossetania.com
cossetania@cossetania.com

Prefacio

Los grandes avances realizados en las tecnologías de la información y de las comunicaciones (TIC) han elevado nuestra capacidad de generar y compartir información hasta límites insospechados, y con esta capacidad también han aumentado los riesgos que ello supone.

El mundo electrónico-digital tiende a reemplazar a los antiguos sistemas, más lentos e ineficientes. Algunos ejemplos cotidianos los encontramos en el correo electrónico, el comercio electrónico, la votación electrónica, la administración digital, la televisión digital, etc. El mundo de lo electrónico y lo digital está llamado a ser el dominante y es por ello que resulta de vital importancia el estudio de teorías y métodos que permitan garantizar la privacidad y la seguridad de los usuarios en este nuevo contexto.

Con esta idea en mente, y con el objetivo de servir de foro de intercambio de conocimientos para los investigadores, en 1991 nació la Reunión Española sobre Criptología y Seguridad de la Información (RECSI), la cual en aquel entonces vino a llamarse Primera Jornada Española sobre Criptografía.

Este año, Tarragona acoge en septiembre la undécima edición de la RECSI, el congreso científico español de referencia en el ámbito de la seguridad en las tecnologías de la información. Las pasadas ediciones se realizaron en Palma de Mallorca (1991), Madrid (1992), Barcelona (1994), Valladolid (1996), Torremolinos (1998), Santa Cruz de Tenerife (2000), Oviedo (2002), Leganés (2004), Barcelona (2006) y Salamanca (2008).

La ciudad de Tarragona ha sido testigo de generaciones cuyas huellas han merecido el reconocimiento mundial. Como elemento representativo de esta RECSI hemos elegido la muralla, símbolo de los vestigios de la Tarragona Romana y uno de los muchos elementos patrimoniales que recomendamos visitar.

Estas actas contienen las 72 contribuciones de la RECSI 2010, cuyas sesiones se organizan en los siguientes ámbitos temáticos: cifrado de flujo, clave pública, criptoanálisis, firmas digitales, privacidad, protocolos, RFID, seguridad, seguridad de redes, watermarking y fingerprinting. Como conferenciantes invitados contamos con Ronald Cramer (Centrum Wiskunde & Informatica, Amsterdam) y Paulo Veríssimo (Universidad de Lisboa).

Desde la organización queremos expresar nuestro agradecimiento a todos los patrocinadores y colaboradores del evento, así como también a todos los ponentes, asistentes, miembros de los comités y revisores.

La compilación de las actas se realizado con LaTeX y el paquete 'confproc'.

Tarragona, septiembre de 2010

Josep Domingo Ferrer
Jordi Castellà Roca
Antoni Martínez Ballesté
Agustí Solanas Gómez

Comité de organización

Presidente

- Josep Domingo Ferrer (Universitat Rovira i Virgili)

Vicepresidentes

- Jordi Castellà Roca (Universitat Rovira i Virgili)
- Antoni Martínez Ballesté (Universitat Rovira i Virgili)
- Agusti Solanas Gómez (Universitat Rovira i Virgili)

Secretaría

- Jesús Manjón Paniagua (Universitat Rovira i Virgili)
- Gloria Pujol Crespo (Universitat Rovira i Virgili)

Comité científico

- Abascal Fuentes, Policarpo (Universidad de Oviedo)
- Álvarez Marañón, Gonzalo (C.S.I.C.)
- Amigó García, José María (Universidad Miguel Hernández)
- Areitio Bertolín, Javier (Universidad de Deusto)
- Borrell Viader, Joan (Universitat Autónoma de Barcelona)
- Bras Amorós, Maria (Universitat Rovira i Virgili)
- Caballero Gil, Pino (Universidad de La Laguna)
- Castellà Roca, Jordi (Universitat Rovira i Virgili)
- Climent, Joan-Josep (Universitat d'Alacant)
- Domingo Ferrer, Josep (Universitat Rovira i Virgili)
- Durán Díaz, Raúl (Universidad de Alcalá de Henares)
- Fernández-Medina Patón, Eduardo (Universidad de Castilla La Macha)
- Ferrer Gomila, Josep Lluís (Universitat de les Illes Balears)
- Fúster Sabater, Amparo (C.S.I.C.)
- González Vasco, Mª Isabel (Universidad Rey Juan Carlos)
- Gutiérrez Gutiérrez, Jaime (Universidad de Cantabria)
- Hernández Encinas, Luis (C.S.I.C.)
- Hernández Goya, Candelaria (Universidad de La Laguna)
- Herrera Joancomartí, Jordi (Universitat Autònoma de Barcelona)
- Huguet Rotger, Llorenç (Universitat de les Illes Balears)

- López Muñoz, Javier (Universidad de Málaga)
- Martín del Rey, Ángel (Universidad de Salamanca)
- Martínez López,Consuelo (Universidad de Oviedo)
- Megías, David (Universitat Oberta de Catalunya)
- Miret Biosca, José María (Universitat de Lleida)
- Morillo Bosch, Paz (Universitat Politècnica de Catalunya)
- Padró Laimon, Carles (Universitat Politècnica de Catalunya)
- Peinado Domínguez, Alberto (Universidad de Málaga)
- Ramió Aguirre, Jorge (Universidad Politécnica de Madrid)
- Ramos Álvarez, Benjamín (Universidad Carlos III de Madrid)
- Ribagorda Garnacho, Arturo (Universidad Carlos III de Madrid)
- Rifà Coma, Josep (Universitat Autònoma de Barcelona)
- Sáez Moreno, Germán (Universitat Politècnica de Catalunya)
- Salazar Riaño, José Luis (Universidad de Zaragoza)
- Sánchez Ávila, Carmen (Universidad Politécnica de Madrid)
- Sebé, Francesc (Universitat de Lleida)
- Sempere Luna, José María (Universitat Politècnica de València)
- Soriano Ibáñez, Miguel (Universitat Politècnica de Catalunya)
- Tena Ayuso, Juan (Universidad de Valladolid)
- Villar Santos, Jorge (Universitat Politècnica de Catalunya)
- Wu, Qianhong (Universitat Rovira i Virgili)
- Zurutuza, Urko (Universidad de Mondragón)

Programa de las sesiones

Cifrado de flujo

Clave pública

Criptoanálisis

Firmas digitales

Privacidad

Protocolos

RFID

Seguridad

Seguridad de redes

Watermarking y fingerprinting

Criptografía de alta velocidad: Cifrando en condiciones extremas (grandes cantidades de datos en tiempo escaso)

Vicente Jara Vera
Carmen Sánchez Ávila
Grupo de Biometría, Bioseñales y Seguridad (GB2S)
Centro de Domótica Integral (CeDInt)
Universidad Politécnica de Madrid, Madrid, España
Email: {vjara,csa}@mat.upm.es

Gonzalo Bailador del Pozo
Javier Guerra Casanova
Alberto de Santos Sierra
Grupo de Biometría, Bioseñales y Seguridad (GB2S)
Centro de Domótica Integral (CeDInt)
Universidad Politécnica de Madrid, Madrid, España
Email: {gbailador,jguerra,alberto}@cedint.upm.es

***Abstract*—Inicialmente al cifrado se le pide seguridad, pero hay entornos en los que además se precisa cifrar una gran cantidad de datos en un tiempo aceptable. La gran cantidad de datos que cada vez se van manejando en las redes actuales, servidores, sistemas de comunicación, puntos de routing, e incluso ordenadores y máquinas individuales hace preciso contar con sistemas que permitan cifrar volúmenes tan grandes como los Gygabytes en tiempos tan cortos o escasos como uno o dos segundos. Ofrecemos en este estudio la situación actual (2010) en cuanto a la existencia, o no, de criptosistemas que permitan realizar esta tarea, así como la sugerencia de cuáles usar en los distintos entornos que pudieran presentársenos.**

I. Introducción

Cifrar gran cantidad de datos en condiciones donde el tiempo es reducido es una situación que en algunos entornos puede ser necesario realizar e incluso hacerlo de manera habitual, lo que hace preciso el disponer de criptosistemas que se adecuen a estas necesidades.

Este es precisamente el caso que estamos considerando en este trabajo, buscar los cifradores más rápidos existentes en la actualidad sin perder, claro esta, la inherente capacidad de seguridad que se espera de un criptosistema, ya que la velocidad en las operaciones del cifrador no ha de ser óbice para rebajar la dificultad de ruptura del sistema criptográfico.

El entorno es sumamente restrictivo, ya que si hablamos de velocidad alta de cifrado estamos pensando en cifrar volúmenes hoy en día considerados grandes (Gigabytes) en tiempos pequeños (segundos). Este intento, ya lo decimos, no tenía solución hace alrededor de 6–8 años, pero hoy sí existen sistemas capaces de lograr tales cotas de eficiencia. Nuestro propósito es mostrar un elenco de la situación actual en base a los mejores criptosistemas existentes en sus distintas variantes (asimétricos y simétricos) para trabajar en situaciones o entornos como el planteado.

II. Criptosistemas simétricos y asimétricos

La división de la Criptografía como una Ciencia puede hacerse en base a dos hechos significativos que son punto de inflexión en la misma. Primeramente, el artículo de 1.948 de Claude Shannon sobre Teoría de la Información y Criptología, "A mathematical Theory of Communications" [1]: es el comienzo de la consideración de la Criptología no ya como un Arte apartado y misterioso, sino como una rama de la Matemática. El segundo acontecimiento es la publicación en el año 1.976 de un artículo por Whitfield Diffie y Martin Hellman, "New Directions in Cryptography" [2], en el que proponen un nuevo sistema de cifrado, el de clave pública. Este segundo artículo supuso la división de los criptosistemas en sistemas de clave secreta o simétrica y sistemas de clave pública o asimétrica.

1) Criptosistemas asimétricos: Los cifradores más rápidos son los cifradores simétricos, ya que los asimétricos descansan en la ejecución de procedimientos y algoritmos matemáticos basados en problemas de complejidad difíciles de resolver, que hacen que sean más lentos en ejecutarse que los simétricos, basados en operaciones matemáticas más rápidas, a veces repetitivas, que añaden confusión y difusión -como especificaba Claude Shannon [1]- a los datos en claro de entrada. Los sistemas simétricos o de clave secreta a su vez pueden catalogarse como cifradores de bloque y de flujo o "stream", según cifren un conjunto de bits a la vez o bien vayan cifrando bit a bit, respectivamente.

No obstante, y debido a que el cifrado de los datos se hace simétricamente, y sólo se usa el cifrado asimétrico para la clave de aquel, no será su lentitud un problema y no serán objeto de nuestro estudio, aunque haremos seguidamente una somera mención de sus capacidades de ejecución.

Así, entre los más conocidos criptosistemas asimétricos y que se consideran seguros, indiquemos que McEliece [3] no siendo lento ofrece unos tamaños de claves sumamente gigantescos; RSA [4] o ElGamal [5], que se basan en el Problema de la Factorización el primero, y en el Problema del Logaritmo Discreto el segundo, son más lentos ([6], [7], [8], [9]), que el cifrado de Curvas Elípticas, basado en el Problema del Logaritmo Discreto en este Grupo algebraico ([10], [11]). Los sistemas Benaloh [12], Goldwasser-Micali [13], Damgaard-Jurik [14], Naccache-Stern [15], Okamoto-Uchiyama [16] o el

de Paillier [17], se basan en el Problema de la Residuosidad Cuadrática o la Compuesta y son como mucho problemas similares al de la Factorización, lo que los convierte en similares a RSA en velocidad por la ejecutoria de las operaciones matemáticas de los algoritmos involucrados; otros sistemas son el de Blum-Goldwasser [18], que a pesar de ser de tipo "stream", no llega a obtener tan buenos resultados en velocidad como el de Curva Elíptica [19]; por otro lado, los que se basan en el Problema del Logaritmo Discreto, y citamos entre otros a CEILIDH [20], Cramer-Shoup [21], y XTR [22], no son tan seguros como el de Curva Elíptica [23] y son siempre inferiores a éste en velocidad ([24], [25]), aunque el último se acerca a él [22].

2) Criptosistemas simétricos en bloque: El criptosistema simétrico en bloque por excelencia hasta la aparición de AES [26] en el año 2.002 ha sido DES [27], el cual es alrededor del doble de lento que el nuevo cifrado simétrico estándar. Siguiendo el único estudio de velocidades realizado sobre los 15 candidatos a mejor cifrado simétrico en los EE.UU por el NIST [28] podemos concluir que si se implementa en software el cifrado en microprocesadores de 32 bits los mejores candidatos para cifrar grandes cantidades de datos son Twofish [29], Rijndael [30] (el original AES, antes de ser redefinido por el estándar), CRYPTON [31], RC6 [32] y MARS [33], y para pequeñas cantidades Rijndael, CRYPTON y E2 [34]. Para procesadores de 64 bits los más veloces son Rijndael, Twofish, DFC [35], MARS y E2. Para el caso de implementaciones hardware, si bien es sumamente complejo de comparar unos con otros por la cantidad de variables constructivas a considerar [28], podemos anotar del mismo artículo que los mejores son CRYPTON, Serpent [36], Twofish, Rijndael y Safer+ [37].

Debido a que los concursos de selección del mejor criptosistema simétrico tanto en EEUU (NIST, 1.997-2.001, [26]), Europa (NESSIE, 2.000-2.003, [38]) y Japón (CRYPTREC, 2.000-2.003, [39]) consideraron siempre al cifrado Rijndael como seleccionado en la serie finalista, podemos considerarlo como un candidato acertado en cuanto a seguridad e implementación, y como hemos podido ver, también el mejor, en general, en referencia a la velocidad de cifrado.

Anotemos, no obstante, que nos estamos moviendo dentro de los órdenes de magnitud de los cifrados simétricos de bloque, en las decenas y centenas de ciclos de reloj por byte cifrado, lo que supone un orden de 10–100 veces más de media de lo que podemos encontrar en los cifrados simétricos de "stream" o de flujo, que serán los que veremos en la siguiente sección, y que constituirán el núcleo más grueso de nuestro estudio, y de entre los cuales acabaremos eligiendo el mejor cifrado para condiciones extremas de velocidad.

III. Criptosistemas simétricos de flujo o de "stream"

Los cifrados de "stream" o de flujo son cifrados simétricos donde el texto claro se combina con la clave, expresada también como un flujo de bits, en general mezclada por la función xor. Este tipo de cifrado toma el texto claro y lo manipula bit a bit (o dígito a dígito, entendiendo "dígito" en sentido amplio) en cada momento, variando el cifrado en función del estado actual del sistema, por lo que también se denomina a este tipo de cifrado "cifrado de estados" (lo que no ocurre jamás en el cifrado en bloques, que es invariante). Intentan asemejarse al cifrado perfecto o de Vernam [40], consistente en un chorro de dígitos aleatorios combinado mediante xor con el texto claro, uno tras otro, el cual fue demostrado ser imposible de romper por Claude Shannon en el año 1.949 [41]. La dificultad en el uso de este sistema está en que la longitud de la clave secreta ha de ser tan larga como el texto a cifrar y que el generador aleatorio ha de serlo realmente. En la realidad los cifrados simétricos de flujo se adaptan a usar claves secretas menores que los de Vernam, como por ejemplo 128 bits pseudoaleatorios. Los criptosistemas de "stream" son más rápidos que los de bloques y presentan una menor complejidad cuando se implementan tanto en hardware -sobretodo- como en software.

Nos centraremos en el siguiente listado -numeralmente muy amplio, lo que nos dará una completa idea de la realidad de este tipo de cifradores- de criptosistemas simétricos de flujo, sobre los que indicaremos su capacidad en eficiencia de velocidad cifrante de altas cantidades de datos:

A5/1: 1.987 [42], A5/2: 1.989 [42], A5/3 (KASUMI): 2.007 [43], BMGL: 2.001 [44], Chamaleon: 1.997 [45], Dragon: 2.005 [46], FISH: 1.993 [47], Grain: 2.004 [48], HC-128: 2.008 [49], HC-256: 2.004 [50], ISAAC: 1.996 [51], Leviathan: 2.000 [52], LILI-128: 2.000 [53], MICKEY: 2.008 [54], MUGI: 2.002 [55], MULTI-s01: 2.002 [56], Panama: 1.998 [57], Phelix: 2.004 [58], Pike: 1.994 [59], Py: 2.005 [60], (TPy, TPypy y TPy6, 2.007, [61]), Rabbit: 2.003 [62], RC4: 1.987 [63], RCR32/64: 2.007 [64], Salsa20: 2.004 [65], Scream: 2.002 [66], SEAL: 1.994 ([67], [68]), SNOW: 2.000 [69], SOBER: 1.997 ([70], [71]), Sosemanuk: 2.005 [72], Trivium: 2.005 [73], Turing: 2.003 [74], VEST: 2.005 [75].

Muchos de ellos fueron presentados al concurso de cifradores de flujo convocados por Europa al descartar NESSIE en el año 2.003 a todos los cifradores presentados. Así pues, Europa creó le Proyecto eSTREAM [76], que entre los años 2.004 y 2.008 seleccionó en el perfil de hardware (para condiciones restrictivas de almacenamiento, potencia y espacio) los criptosistemas Grain v.1, MICKEY v.2 y Trivium. En el perfil de software los elegidos fueron HC-128, Rabbit, Salsa20/12 (es la versión Salsa 12 con 12 rondas) y Sosemanuk. Indiquemos que los cifradores de software tienden a ser más veloces que los de hardware, si bien cuando se implementan en software estos últimos resultan ser todavía más lentos que sus implementaciones hardware; siempre hay excepciones como Trivium (cifrador hardware), entre otros, comparativamente similar en velocidad a los más rápidos de los cifrados software ([77], [73]).

Tras repasar la anterior lista de cifrados, si quitamos a la hora de optar por los que podrían sernos de interés, los orientados a hardware elegidos por eSTREAM (Grain, MICKEY, Trivium, VEST), los que son versiones anteriores de otros que

los mejoran (SEAL) [66], los multicanales, que no es el caso que nos ocupa por no ser de propósito general (HC-128, HC-256), los que se usan para la protección de derechos de autor y usan de marcados, que lo ralentizan para nuestros propósitos (Chamaleon), y los que no son seguros, así considerados hoy en día (año 2.010), aunque a veces solamente a nivel teórico, (A5/1 ([78], [79]), A5/2 [80], A5/3 [81], Dragon [82], FISH [59], Leviathan [83], LILI-128 [84], Phelix [85], Py ([86], [85]), TPy ([86], [87], [85], [88], [89]), TPy6 ([90], [85]), RC4 ([91], [92]), Trivium ([93], [94], [95], [96]), Turing [97]), nuestra lista de los mejores representantes de cifrados de flujo, dando la referencia de la velocidad de cada uno de ellos, queda de la siguiente manera:

BMGL ([44], [52]), ISAAC [98], MUGI [99], Multi-s01 [100], Panama [57], Pike [59], Rabbit [62], RCR-32 [64], RCR-64 [64], Salsa20/12 [101], Scream [66], SNOW2 [69], SOBER [102], Sosemanuk [72], TPypy [64].

Si ahora los clasificamos anotando su velocidad en ciclos de reloj por byte y si han sido o no elegidos en los dos Proyectos de cifradores de flujo, CRYPTREC o eSTREAM, tenemos el siguiente gráfico de la figura 1 y 2.

	Velocidad (ciclos/byte)	Elegido en la final software de eSTREAM	Elegido en CRYPTREC
BMGL	5		
ISAAC	4,6		
MUGI	21,8		Elegido
Multi-s01	17,7		Elegido
Panama	4,7		
Pike	20		
Rabbit	3,7	Elegido	
RCR-32	2,7		
RCR-64	4,45		
Salsa20/12	13	Elegido	
Scream	4,5		
SNOW2	5,5		
SOBER	5		
Sosemanuk	5	Elegido	
TPypy	4,58		

Fig. 1. Cifradores de flujo más rápidos existentes hoy en día.

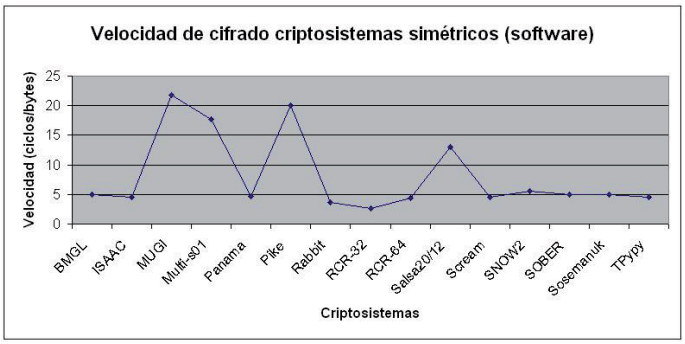


Fig. 2. Representación comparativa de los cifradores mostrados en la Fig. 1.

Si tomamos de la lista los mejor situados y analizamos además el Benchmarck del europeo ECRYPT II [103], podemos sacar algunas tablas adicionales de interés para nuestra búsqueda que ya nos dan valores de cifrado del entorno de los Gigabytes de datos por segundo, como muestra la figura 3.

IA 64, 997MHz, HP Itanium II	Velocidad media (ciclos/byte)
Salsa20/12	11,98
TPypy	15,87

PPC 32, 1900MHz, IBM Power 5	Velocidad media (ciclos/byte)
TPypy	5,13
Salsa20/12	7,04
Sosemanuk	7,34
Rabbit	9,15

AMD 64, 3192MHz, Intel Pentium 4	Velocidad media (ciclos/byte)
SNOW2	5,24
Sosemanuk	6,87
Salsa20/12	7,03
TPypy	8,58
Rabbit	20,00

IA 64, 1396MHz, HP Itanium II	Velocidad media (ciclos/byte)
Salsa20/12	13,00
TPypy	17,00

AMD 64, 2300MHz, AMD Opteron 2376	Velocidad media (ciclos/byte)
Sosemanuk	2,64
Panama	2,93
Salsa20/12	3,29
SNOW2	3,86
Rabbit	4,81
TPypy	5,87
AES128-CTR	9,86

AMD 64, 2992MHz, Intel Pentium D	Velocidad media (ciclos/byte)
SNOW2	4,97
Salsa20/12	6,78
Sosemanuk	6,83
Panama	9,02
TPypy	9,16
Rabbit	10,06

IV. Conclusiones

De entre todos los cifradores simétricos, tanto de bloque como de flujo, recordar que los últimos son de orden de 10 a 100 veces superiores en prestaciones de velocidad a los de bloque, aunque algunos de los más rápidos de entre éstos como Blowfish están en el orden de los 18 ciclos/byte [104], o AES-128 en modo CTR alcanza en el Opteron, como hemos visto en la figura 3, un valor de 9,86 ciclos/byte.

No obstante, tras nuestro estudio, hemos de concluir que la elección final para cifradores simétricos -y cifradores de cualquier tipo, en definitiva, simétricos o asimétricos- está en usar cifradores de flujo o “stream”, tomando en cada caso

AMD 64, 2400MHz, Intel Core 2 Quad Q6600	Velocidad media (ciclos/byte)
Panama	2,03
Sosemanuk	2,40
Salsa20/12	2,65
SNOW2	4,30
TPypy	4,38
Rabbit	5,44

X86, 2833MHz, Intel Core 2 Quad Q9550	Velocidad media (ciclos/byte)
Panama	1,95
Sosemanuk	2,29
Rabbit	2,37
Salsa10/12	2,86
TPypy	4,20
SNOW2	4,32

Fig. 3. Algunas tablas comparativas de los más rápidos cifradores en diferentes procesadores.

particular, sobre la lista final mostrada en la figura 1, el que más se adecue a las características de procesador de la aplicación considerada en cada escenario, con lo que obtendremos unos valores de cifrado del orden de los Gigabytes de datos por segundo en un ordenador usuario medio, cumpliendo de manera adecuada los requerimietos de poder cifrar cantidades enormes, del orden de los Gygabytes, en tiempos tan cortos como un segundo, algo impensable hace sólo 6 ó 8 años.

REFERENCES

[1] C. E. Shannon, "A Mathematical Theory of Communication", en *B*ell System Technical Journal, 27, pp.379–423, 1948.

[2] W. Diffie, M. E. Hellman, "New Directions in Cryptography", en *I*EEE Transactions on Information Theory, vol. 22, no. 6, pp.644–654, 1976.

[3] R. J. McEliece, "A Public-Key Cryptosystem Based On Algebraic Coding Theory", en *D*SN Progress Report, pp.42–44, 1978.

[4] R. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", en *C*ommunications of the ACM, vol. 21, no. 2, pp.120–126, 1978.

[5] T. ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", en *I*EEE Transactions on Information Theory, vol. 31, no. 4, pp.469–472, 1985.

[6] B. Kalisky, "TWIRL and RSA Key Size", en *A*vailable online: http://www.rsa.com/rsalabs/node.asp?id=2004, 2003.

[7] NIST, "The Case for Elliptic Curve Cryptography", en *A*vailable online: http://www.nsa.gov/business/programs/elliptic_curve.shtml, 2009.

[8] V. Gupta, D. Stebila, S. Fung, S. Ch. Shantz, N. Gura, H. Eberle, "Speeding Up Secure Web Transactions Using Elliptic Curve Cryptography", en *N*DSS, 2004.

[9] Certicom, *A*vailable online: https://www.certicom.com/index.php?action=company,press_archive&view=307.

[10] V. Miller, "Use of Elliptic Curves in Cryptography", en *C*RYPTO 85, pp.417–426, 1985.

[11] N. Koblitz, "Elliptic Curve Cryptosystems", en *M*athematics of Computation, vol. 48, pp.203–209, 1987.

[12] J. Benaloh, "Dense Probabilistic Encryption´´ en *P*roceedings. SAC '9# - Workshop on Selected Areas of Cryptography, 1994.

[13] S. Goldwasser. S. Micali, "Probabilistic Encryption", en *J*ournal of Computer and System Sciences, vol. 28, no. 2, pp.270–299, 1984.

[14] I. Damgard, M. Jurik, "A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System", en *P*ublic Key Cryptography 2001, pp.119–136, 2001.

[15] D. Naccache , J. Stern, "A New Public Key Cryptosystem Based on Higher Residues", en *P*roceedings of the 5th ACM Conference on Computer and Communications Security, pp.59–66, 1998.

[16] T. Okamoto, S. Uchiyama, "A New Public-Key Cryptosystem as Secure as Factoring", en *A*dvances in Cryptology, pp.308–318, 1998.

[17] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes", en *E*UROCRYPT 1999, pp.223–238, 1999.

[18] M. Blum, S. Goldwasser, "An Efficient Probabilistic Public Key Encryption Scheme which Hides All Partial Information", en *P*roceedings of Advances in Cryptology - CRYPTO 84, pp.289–299, 1985.

[19] R. Shanmugalakshmi, M. Prabu, "Research Issues on Elliptic Curve Cryptography and Its applications", en *I*nternational Journal of Computer Sciencie and Network Security, vol. 9, no. 6, pp.19–22, 2005.

[20] K. Rubin, A. Silverberg, "Torus-Based Cryptography", en *C*RYPTO 03, pp.349–365, 2003.

[21] R. Cramer, V. Shoup, "A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack", en *P*roceedings of Crypto LNCS, 1998.

[22] A. K. Lenstra, E. R. Verheul, "An Overview of the XTR Public Key System", en *L*ectures. Notes in Computer Sciences, pp.1–19. 2000.

[23] V. Jara Vera, C. Sánchez Ávila, "Análisis comparativo entre métodos de ataque a los criptosistemas RSA, ElGamal y Curvas Elípticas", en *I*I Simposio sobre Seguridad Informática, CEDI 07, 2007.

[24] B. Schneier, "Applied Cryptography", Second Edition, Wiley, 1996.

[25] L. Ertaul, N. Chavan, "RSA and Elliptic Curve- ElGamal Threshold Cryptography (ECCEG-TC) Implementations for Secure Data Forwarding in MANETs", en *S*ecurity and Management, pp.142–146, 2007.

[26] NIST, "Advanced Encryption Standard", en *F*IPS PUB 197, National Bureau of Standards, U.S. Department of Commerce, 2001.

[27] NIST, "Data Encryption Standard", en *F*IPS PUB 46, National Bureau of Standards, U.S. Department of Commerce, 1977.

[28] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, Ch. Hall, N. Ferguson, "Performance Comparison of the AES Submissions", en *P*roceedings Second AES Candidate Conference, NIST, pp.15–34, 1999.

[29] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, Ch. Hall, N. Ferguson, "The Twofish Encryption Algorithm", 1998.

[30] L. R. Knudsen, V. Rijmen, R. L. Rivest, M. J. B. Robshaw, "On the Design and Security of RC2", en *F*ast Software Encryption, pp.206–221. 1998.

[31] E. Hong, J. Chung, "Hardware Design and Performance Estimation of The 128-bit Block Cipher CRYPTON", en *P*roceedings of the First International Workshop on Cryptographic Hardware and Embedded Systems, 1717, pp.49–60, 1999.

[32] R. L. Rivest, M. J. B. Robshaw, R. Sidney, Y. L. Yin, "The RC6 Block Cipher", 1998.

[33] C. Burwick, D. Coppersmith, E. D´Avignon, R. Gennaro, S. Halevi, Ch. Jutla, S. M. Matyas Jr., L. O´Connor, M. Peyravian, D. Safford, N. Zunic, "MARS: A Candidate Cipher for AES", en *P*roceedings from the First Advanced Encryption Standard Candidate Conference, NIST, 1998.

[34] K. Masayuki, M. Shiho, A. Kazumaru, U. Hiroki, "The E2 Block Cipher (AES Proposed Version)", en *N*TT R D, vol. 48, no. 10, pp.723–733, 1999.

[35] H. Gilbert, M. Girault, P. Hoogvorst, F. Noilhan, T. Pornin, G. Poupard, J. Stern, S. Vaudenay, "Decorrelated Fast Cipher: an AES candidate", en *P*roceedings from the First Advanced Encryption Standard Candidate Conference, NIST, 1998.

[36] R. Anderson, E. Biham, L. Knudsen, "Serpent: A Candidate Block Cipher for the Advanced Encryption Standard", en *P*roceedings from the First Advanced Encryption Standard Candidate Conference, NIST, 1998.

[37] J. L. Massey, "SAFER K-64: A Byte-Oriented Block-Ciphering Algorithm", en *F*ast Software Encryption, pp.1–17, 1993.

[38] NESSIE, New European Schemes for Signatures, Integrity, and Encryption, IST-1999-12324, en *A*vailable online: https://www.cosic.esat.kuleuven.be/nessie/, 2003.

[39] CRYPTREC, *A*vailable online: http://www.cryptrec.go.jp/english, 2003.

[40] G. S. Vernam, "Secret Signaling System", en *U*S Patent 1310719, 1919.

[41] C. Shannon, "Communication Theory of Secrecy Systems", en *B*ell System Technical Journal, vol. 28, no. 4, pp.656–715, 1949.

[42] M. Briceno, I. Goldberg, D. Wagner, "A Pedagogical Implementation of the GSM A5/1 and A5/2 "Voice Privacy" Encryption Algorithms", 1998.

[43] ETSI, "Universal Mobile Telecommunications System (UMTS)", en *S*pecification of the 3GPP confidentiality and integrity algorithms, Document 2: Kasumi specification, 2007.

[44] J. Hastad, M. Naslund, "BMGL, Synchronous Key–stream Generator with Provable Security", en *P*roject NESSIE, 2000.

[45] R. Anderson, Ch. Manifavas, "Chamaleon, A New Kind of Stream Cipher", en *F*ast Software Encryption, vol. 1267, pp.107–113, 1997.

[46] E. Dawson, K. Chen, M. Henricksen, W. Millan, L. Simpson, H. Lee, S. Moon, "Dragon: A Fast Word Based Stream Cipher Dragon", en *P*roject eSTREAM, 2005.

[47] U. Blöcher, M. Dichtl, "Fish: A Fast Software Stream Cipher", en *F*ast Software Encryption, pp.41–44, 1993.

[48] M. Hell, T. Johansson, W. Meier, "Grain, A Stream Cipher for Constrained Environments", en *P*roject eSTREAM, 2004.

[49] H. Wu, "The Stream Cipher HC-128", en *P*roject eSTREAM, 2008.

[50] H. Wu, "Stream Cipher HC-256", en *P*roject eSTREAM, 2004.

[51] R. J. Jenkins Jr., "ISAAC", en *F*ast Software Encryption, pp.41–49, 1996.

[52] D. A. McGrew, S. R. Fluhrer, "The Stream Cipher LEVIATHAN: Specification and Supporting Documentation", en *P*roject NESSIE, 2000.

[53] E. Dawson, A. Clark, J. Golic, W. Millan, L. Penna, L. Simpson, "The LILI–128 Keystream Generator", en *P*roject NESSIE, 2000.

[54] S. Babbage, M. Dodd, "The MICKEY stream ciphers", en *L*ecture Notes in Computer Science, vol. 4986, pp.191–209, 2008.

[55] D. Watanabe, S. Furuya, K. Takaragi, B. Preneel, "A New Keystream Generator MUGI", en *F*ast Software Encryption, pp.179–194, 2002.

[56] S. Furuya, K. Sakurai, "Single–path Authenticated–encryption Scheme Based on Universal Hashing", en *S*elected Areas in Cryptography, 9th Annual Workshop, SAC 02, vol. 2595, pp.94–109, 2002.

[57] J. Daemen, C. S. K. Clapp, "Fast Hashing and Stream Encryption with PANAMA", en *F*ast Software Encryption, pp.60–74, 1998.

[58] D. Whiting, B. Schneier, S. Lucks, F. Muller, "Phelix: Fast Encryption and Authentication in a Single Cryptographic Primitive", en *P*roject eSTREAM, 2003.

[59] R. Anderson, "On Fibonacci Keystream Generators", en *F*ast Software Encryption, pp.346–352, 1994.

[60] E. Biham, J. Seberry, "Py Roo: A Fast and Secure Stream Cipher using Rolling Arrays", en *P*roject eSTREAM, 2005.

[61] E. Biham, J. Seberry, "Tweaking the IV Setup of the Py Family of Stream Ciphers, The Ciphers TPy, TPypy, and TPy6", en *P*roject eSTREAM, 2007.

[62] M. Boesgaard, M. Vesterager, T. Pedersen, J. Christiansen, O. Scavenius, "Rabbit: A High–Performance Stream Cipher", en *F*ast Software Encryption, pp.307–329, 2003.

[63] H. Finney, "An RC4 cycle that Can't Happen", en *s*ci.crypt, 1994.

[64] S. Gautham, P. Souradyuti, B. Preneel, "Related–key Attacks on the Py–family of Ciphers and an Approach to Repair the Weaknesses", en *I*ndocrypt 07, 2007.

[65] D. J. Bernstein, "The Salsa20 Family of Stream Ciphers", en *P*roject eSTREAM, 2007.

[66] D. Coppersmith, S. Halevi, C. Jutla, "Scream: A Software–Efficient Stream Cipher", en *F*ast Software Encryption, pp.195–209, 2002.

[67] P. Rogaway, D. Coppersmith, "Software–efficient Pseudorandom Function and the Use Thereof for Encryption", en *U*.S. Patent 5,454,039, 1994.

[68] P. Rogaway, D. Coppersmith, "Computer Readable Device Implementing a Software–efficient Pseudorandom Function Encryption", en *U*.S. Patent 5,675,652, 1997.

[69] P. Ekdahl, T. Johansson, "SNOW, a New Stream Cipher", en *P*roceedings of first NESSIE Workshop, 2000.

[70] P. Hawkes, G. Rose, "Primitive Specification and Supporting Documentation for SOBER–t16 Submission to NESSIE", en *P*roceedings of first NESSIE Workshop, 2000.

[71] P. Hawkes, G. Rose, "Primitive Specification and Supporting Documentation for SOBER–t32 Submission to NESSIE", en *P*roceedings of first NESSIE Workshop, 2000.

[72] C. Berbain, O. Billet, A. Canteaut, N. Courtios, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Ptonin, H. Sibert, "Sosemanuk, a Fast Software–oriented Stream Cipher", en *P*roject eSTREAM, 2005.

[73] Ch. De Cannière, B. Preneel, "Trivium, A Stream Cipher Construction Inspired by Block Cipher Design Principles", en *P*roject eSTREAM, 2005.

[74] G. G. Rose, P. Hawkes, "Turing: A Fast Stream Cipher", en *F*ast Software Encryption, pp.290–306, 2003.

[75] S. O'Neil, B. Gittins, H. Landman, "VEST, Hardware–Dedicated Stream Ciphers", en *P*roject eSTREAM, 2005.

[76] ECRYPT, "The eSTREAM Project" *A*vailable online: http://www.ecrypt.eu.org/stream/, 2008.

[77] D. Gligoroski, S. Markovski, S. J. Knapskog, "The Stream Cipher Edon80", en *L*ecture Notes in Computer Science, vol. 4986, pp.152–169, 2008.

[78] A. Maximov, T. Johansson, S. Babbage, "An Improved Correlation Attack on A5/1", en *S*elected Areas in Cryptography, pp. 1–18, 2004.

[79] E. Barkan, E. Biham, "Conditional Estimators: An Effective Attack on A5/1", en *S*elected Areas in Cryptography, pp.1–19, 2005.

[80] E. Barkan, E. Biham, N. Keller, "Instant Ciphertext-only Cryptanalysis of GSM Encrypted Communication", en *J*ournal of Cryptology, vol. 21, no. 3, pp.392–429, 2008.

[81] O. Dunkelman, N. Keller, A. Shamir, "A Practical-Time Attack on the A5/3 Cryptosystem Used in Third Generation GSM Telephony", en *I*ACR-ePrint report, 2010.

[82] J. Y. Cho, J. Pieprzyk, "An Improved Distinguisher for Dragon", en *C*oding Theory and Cryptology, Proceedings of the First International Workshop Wuyi Mountain, Fujian, China, vol. 4, pp.249–265, 2007.

[83] P. Crowley, S. Lucks, "Bias in the LEVIATHAN stream cipher", en *F*ast Software Encryption, 8th International Workshop Yokohama, Japan, 2001.

[84] X. Huang, W. Huang, X. Liu, C. Wang, Z. J. Wang, T. Wang, "Reconstructing the Nonlinear Filter Function of LILI-128 Stream Cipher Based on Complexity", en *A*vailable online: http://arxiv.org/ftp/cs/papers/0702/0702128.pdf, 2007.

[85] H. Wu, B. Preneel, "Differential-Linear Attacks against the Stream Cipher Phelix", en *I*4th International Workshop on Fast Software Encryption, Luxembourg, 2007.

[86] H. Wu, B. Preneel, "Key Recovery Attack on Py and Pypy with Chosen IVs", en *P*roject eSTREAM, 2006.

[87] G. Sekar, S. Paul, B. Preneel, "Weaknesses in the Pseudorandom Bit Generation Algorithms of the Stream Ciphers TPypy and TPy", en *I*ACR-ePrint report, 2007.

[88] H. Wu, B. Preneel, "Differential Cryptanalysis of the Stream Ciphers Py, Py6 and Pypy", en *A*dvances in Cryptology - EUROCRYPT 2007, vol. 4515, pp.276–290, 2007.

[89] P. Crowley, "Improved Cryptanalysis of Py", en *A*vailable online: http://www.ecrypt. eu.org/stream/papersdir/2006/010.pdf, 2006.

[90] G. Sekar, S. Paul, B. Preneel, "New Attacks on the Stream Cipher TPy6 and Design of New Ciphers the TPy6-A and the TPy6-B", en *R*esearch in Cryptology: Second Western European Workshop, WEWoRC 2007, Bochum, Germany, pp.127–141, 2007.

[91] A. Klein, "Attacks on the RC4 stream cipher" en *D*esigns, Codes and Cryptography, vol. 48, pp.269–286, 2008.

[92] E. Tews, R. P. Weinmann, A. Pyshkin, "Breaking 104-bit WEP in under a minute", en *I*ACR-ePrint report, 2007.

[93] I. Dinur, A. Shamir, "Cube Attacks on Tweakable Black Box Polynomials", en *P*roceedings of the 28th Annual International Conference on Advances in Cryptology: the Theory and Applications of Cryptographic Techniques - EUROCRYPT 2009, vol. 5479, pp.278–299, 2009.

[94] M. Vielhaber, "Breaking ONE.FIVIUM by AIDA an Algebraic IV Differential Attack", en *I*ACR-ePrint report, 2007.

[95] M. Vielhaber, "Shamir's cube attack: A Remake of AIDA, The Algebraic IV Differential Attack", en *A*vailable online: http://hs-bremerhaven.de/Binaries/Binary10017/AIDA_Shamir.pdf?SID=846650329488c36272d2aa88efcd870b, 2009.

[96] A. Maximov, A. Biryukov, "Two Trivial Attacks on Trivium". en *S*elected Areas in Cryptography, pp.36–55, 2007.

[97] A. Joux, F. Muller, "A Chosen IV Attack Against Turing" en *S*elected Areas in Cryptography, pp.194–207, 2003.

[98] R. J. Jenkins Jr., "ISAAC: a fast cryptographic random number generator", en *A*vailable online: http://burtleburtle.net/bob/rand/isaacafa.html, 2008.

[99] HITACHI, "MUGI Pseudorandom Number Generator. Self-evaluation Report, v1.1", en *T*echnical Report, 2001.

[100] HITACHI, "MULTI-S01. Self-evaluation Report, Technical Report", 2001.

[101] D. J. Bernstein, "Salsa20. Software Speed", en *A*vailable online: http://cr.yp.to/salsa20/speed.html, 2008.

[102] P. Hawkes, M. Paddon, G. G. Rose, "Primitive Specification for SOBER-128", en *T*echnical Report, Qualcomm Australia, 2003.

[103] ECRYPT II, "eBACS: ECRYPT Benchmarking of Cryptographic Systems", en *A*vailable online: http://bench.cr.yp.to/results-stream.html, 2010.

[104] B. Schneier, "Speed Comparisons of Block Ciphers on a Pentium", en *A*vailable online: http://www.schneier.com/blowfish-speed.html, 2007.

Cálculo del grado de una función booleana a partir de su soporte

Joan-Josep Climent
Departament d'Estadística i Investigació Operativa. Universitat d'Alacant
Email: jcliment@ua.es

Francisco J. García
Departament de Fonaments de l'Anàlisi Econòmica. Universitat d'Alacant
Email: francisco.garcia@ua.es

Verónica Requena
Departament d'Estadística i Investigació Operativa. Universitat d'Alacant
Email: jcliment@ua.es

***Resumen*—En este artículo establecemos algunas propiedades de la forma normal algebraica de una función booleana a partir de su soporte y proponemos un método para determinar el grado de la misma a partir del soporte.**

I. Introducción

Las funciones booleanas se utilizan en distintas aplicaciones criptográficas tales como cifrado en bloque, cifrado en flujo y funciones hash [2], [4], [7], [9] y en teoría de códigos [3], [6], entre otras. Las propiedades más importantes de las funciones booleanas son la no linealidad, la inmunidad a la correlación y su grado. Por ejemplo, la implementación de una caja de sustitución o S-box necesita funciones booleanas no lineales que garanticen su efectividad criptográfica con el fin de resistir ataques tales como el criptoanálisis lineal o el criptoanálisis diferencial [1], [5], [8], [10].

Una de las exigencias más básica relativa a las funciones booleanas utilizadas en cifradores en flujo es que permitan incrementar la complejidad lineal [9], [15], [16], lo cual se consigue si éstas tienen un alto grado algebraico.

La determinación completa de la forma normal algebraica de una función booleana de la que se conoce su tabla de verdad o su soporte requiere computar simultáneamente todos los coeficientes del polinomio correspondiente, pero si se desea conocer solamente el grado de la función, es posible reducir sustancialmente el número de operaciones necesarias mediante el algoritmo que presentamos aquí.

El resto del artículo está organizado como sigue. En la sección II introducimos algunos conceptos básicos y la notación que utilizaremos a lo largo del artículo y en la sección III introducimos los resultados principales del artículo; en particular, introducimos una serie de propiedades que nos permitirán determinar el grado de una función booleana de n variables a partir de su soporte. Finalizamos con las conclusiones y líneas futuras en la sección IV.

II. Preliminares

Denotamos por $\mathbb{F}_2$ el cuerpo de Galois de dos elementos 0 y 1 con la adición (denotada por $\oplus$) y la multiplicación (denotada por yuxtaposición). Para cada entero positivo n, es bien conocido que $\mathbb{F}_2^n$ es un espacio vectorial de dimensión n sobre $\mathbb{F}_2$ con la adición usual (denotada también por $\oplus$). Denotamos por $\mathrm{Env}\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k\}$ el subespacio vectorial de $\mathbb{F}_2^n$ generado por los vectores $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k \in \mathbb{F}_2^n$. Si $\boldsymbol{u} = (u_1, u_2, \ldots, u_n) \in \mathbb{F}_2^n$, llamamos **peso** de $\boldsymbol{u}$, y escribimos $w(\boldsymbol{u})$, al número de componentes iguales a 1 y si consideramos 0 y 1 como elementos de $\mathbb{Z}$, claramente $w(\boldsymbol{u}) = \sum_{j=1}^{n} u_j$. Si para $i = 0, 1, 2, \ldots, 2^n - 1$, denotamos por $\boldsymbol{i}$ la expansión binaria de i de n dígitos, entonces $\mathbb{F}_2^n = \{\boldsymbol{i} \mid 0 \leq i \leq 2^n - 1\}$.

Una **función booleana** de n variables es una aplicación $f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$. Denotamos por $\mathcal{B}_n$ el conjunto de todas las funciones booleanas de n variables; es bien conocido que $\mathcal{B}_n$, con la adición usual de funciones (que seguimos denotando por $\oplus$), es un espacio vectorial de dimensión 2^n sobre $\mathbb{F}_2$.

Si $f \in \mathcal{B}_n$, llamamos **tabla de verdad** de f (véase, por ejemplo [11], [12]) a la secuencia binaria de longitud 2^n dada por

$$\boldsymbol{\xi}_f = (f(\mathbf{0}), f(\mathbf{1}), \ldots, f(\mathbf{2^n - 1})).$$

Llamamos **soporte** de f, y escribimos $\mathrm{Sop}\,(f)$, al conjunto de vectores de $\mathbb{F}_2^n$ cuya imagen por f es 1, es decir

$$\mathrm{Sop}\,(f) = \{\boldsymbol{a} \in \mathbb{F}_2^n \mid f(\boldsymbol{a}) = 1\}.$$

Si $f \in \mathcal{B}_n$, llamamos **peso** de f, y escribimos $w(f)$, al número de 1 de la tabla de verdad de f, por tanto, $w(f) = |\mathrm{Sop}\,(f)|$ y claramente

$$w(f) = \sum_{\boldsymbol{a} \in \mathbb{F}_2^n} f(\boldsymbol{a}).$$

Decimos que f es **equilibrada** si $w(f) = 2^{n-1}$.

Obviamente f es la función nula si y sólo si $\mathrm{Sop}\,(f) = \emptyset$ y entonces $w(f) = 0$. Análogamente f es la función constante 1 si y sólo si $\mathrm{Sop}\,(f) = \mathbb{F}_2^n$ y en tal caso $w(f) = 2^n$.

Es fácil comprobar que si $f, g \in \mathcal{B}_n$, entonces

$$\mathrm{Sop}\,(f \oplus g) = \mathrm{Sop}\,(f)\,\Delta\,\mathrm{Sop}\,(g)$$

donde Δ denota de diferencia simétrica de conjuntos y, en consecuencia,

$$w(f \oplus g) \equiv w(f) + w(g) \pmod 2.$$

En general, si $f_j \in \mathcal{B}_n$, para $j = 1, 2, \ldots, m$, entonces

$$\mathrm{Sop}\left(\bigoplus_{j=1}^{m} f_j\right) = \bigtriangleup_{j=1}^{m} \mathrm{Sop}\,(f_j) \tag{1}$$

y, por tanto,

$$w\left(\bigoplus_{j=1}^{m} f_j\right) \equiv \sum_{j=1}^{m} w(f_j) \pmod 2.$$

Supongamos ahora que $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ donde cada x_j, para $j = 1, 2, \ldots, n$, es una variable binaria. Si $f \in \mathcal{B}_n$, podemos escribir $f(\boldsymbol{x})$ de forma única (véase, por ejemplo, [6], [11], [12], [13], [14]) como

$$f(\boldsymbol{x}) = \bigoplus_{\boldsymbol{u}\in\mathbb{F}_2^n} \mu_f(\boldsymbol{u})\boldsymbol{x}^{\boldsymbol{u}} \tag{2}$$

donde $\mu_f(\boldsymbol{u}) \in \mathbb{F}_2$ y si $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$, entonces

$$\boldsymbol{x}^{\boldsymbol{u}} = x_1^{u_1} x_2^{u_2} \cdots x_n^{u_n} \quad \text{con} \quad x_j^{u_j} = \begin{cases} x_j, & \text{si } u_j = 1, \\ 1, & \text{si } u_j = 0. \end{cases}$$

La expresión (2), en la que cada uno de los términos $\boldsymbol{x}^{\boldsymbol{u}}$ es un **monomio** cuyo grado es $w(\boldsymbol{u})$, se conoce con el nombre de **forma normal algebraica** (FNA) de $f(\boldsymbol{x})$. Llamamos **grado** de f, y escribimos $\operatorname{gr}(f)$, al mayor de los grados de los monomios de su FNA, es decir

$$\operatorname{gr}(f) = \max\{w(\boldsymbol{u}) \mid \mu_f(\boldsymbol{u}) = 1\}.$$

Decimos que f es una **función afín** si $\operatorname{gr}(f) = 1$; en tal caso, la FNA de f es

$$f(\boldsymbol{x}) = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus \cdots \oplus a_nx_n$$

con $a_j \in \mathbb{F}_2$, para $j = 0, 1, 2, \ldots, n$. En particular, si $a_0 = 0$, decimos que f es una **función lineal**. Es fácil probar que toda función afín es equilibrada, aunque el recíproco no es cierto.

Por otro lado, si

$$\boldsymbol{\mu}_f = (\mu_f(\mathbf{0}), \mu_f(\mathbf{1}), \ldots, \mu_f(\mathbf{2^n - 1})),$$

entonces (véase por ejemplo [13])

$$\boldsymbol{\mu}_f = \boldsymbol{\xi}_f A_n$$

donde

$$A_n = \begin{bmatrix} A_{n-1} & A_{n-1} \\ O & A_{n-1} \end{bmatrix} \text{ para } n \geq 1, \quad \text{con } A_0 = [1].$$

Ahora, si

$$\begin{aligned} \boldsymbol{u} &= (u_1, u_2, \ldots, u_n) \\ &= u_1\mathbf{2^{n-1}} \oplus u_2\mathbf{2^{n-2}} \oplus \cdots \oplus u_{n-1}\mathbf{2} \oplus u_n\mathbf{1} \end{aligned}$$

y $E(\boldsymbol{u}) = \operatorname{Env}\left\{u_1\mathbf{2^{n-1}}, u_2\mathbf{2^{n-2}}, \ldots, u_{n-1}\mathbf{2}, u_n\mathbf{1}\right\}$, entonces es fácil probar por inducción sobre n, que

$$\mu_f(\boldsymbol{u}) = \bigoplus_{\boldsymbol{a}\in E(\boldsymbol{u})} f(\boldsymbol{a}). \tag{3}$$

III. Resultados principales

En todo este artículo denotamos por S_n, como es habitual, el conjunto formado por las permutaciones de $\{1, 2, \ldots, n\}$. Además, si $\sigma \in S_n$, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$, $M \subseteq \mathbb{Z}_2^n$ y $f \in \mathcal{B}_n$, escribimos $\boldsymbol{x}^\sigma = \left(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)}\right)$,

$$M^\sigma = \{\boldsymbol{x}^\sigma \mid \boldsymbol{x} \in M\} \quad \text{y} \quad f^\sigma(\boldsymbol{x}) = f(\boldsymbol{x}^\sigma).$$

El resultado siguiente, cuya demostración es inmediata, establece la relación entre $\operatorname{Sop}(f^\sigma)$ y $\operatorname{Sop}(f)$ para todo $\sigma \in S_n$.

Teorema 1: *Supongamos que $\sigma \in S_n$. Si $f \in \mathcal{B}_n$, entonces*

$$\operatorname{Sop}(f^\sigma) = (\operatorname{Sop}(f))^{\sigma^{-1}}$$

y, en consecuencia, $w(f^\sigma) = w(f)$.

El resultado siguiente, cuya demostración también es inmediata, permite determinar, de forma explícita, el soporte y, por tanto, el peso, de cualquier monomio.

Teorema 2: *Supongamos que $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ y consideremos $\sigma \in S_n$ tal que $\sigma(j) = i_j$, para $j = 1, 2, \ldots, k$. Si $f(\boldsymbol{x}) = x_{i_1}x_{i_2}\cdots x_{i_k}$ y $\boldsymbol{u} = (1, 1, \ldots, 1) \in \mathbb{F}_2^k$, entonces*

$$(\operatorname{Sop}(f))^\sigma = \{\boldsymbol{u}\} \times \mathbb{F}_2^{n-k} \tag{4}$$

y, en particular, $w(f) = 2^{n-k}$.

Una consecuencia inmediata del resultado anterior es que el peso del monomio formado por todas las variables (es decir, cuando $k = n$) es 1, mientras que el peso de cualquier otro monomio es una potencia de 2 y, por tanto, un número par.

Otra consecuencia inmediata del teorema 2 es que si el grado de una función booleana de n variables es menor o igual que $n-2$, entonces la suma de los elementos de su soporte es el vector nulo. Antes, sin embargo, introducimos el siguiente lema técnico que nos permitirá simplificar la demostración de dicho resultado.

Lema 1: *Supongamos que $1 \leq i_1 < i_2 < \cdots < i_k \leq n$. Si $f(\boldsymbol{x}) = x_{i_1}x_{i_2}\cdots x_{i_k}$ y $1 \leq k \leq n-2$, entonces*

$$\bigoplus_{\boldsymbol{a}\in\operatorname{Sop}(f)} \boldsymbol{a} = \mathbf{0}.$$

Demostración: Supongamos que $\boldsymbol{u} = (1, 1, \ldots, 1) \in \mathbb{F}_2^k$. Claramente

$$\begin{aligned} \bigoplus_{\boldsymbol{a}\in\{\boldsymbol{u}\}\times\mathbb{F}_2^{n-k}} \boldsymbol{a} &= \bigoplus_{\boldsymbol{v}\in\mathbb{F}_2^{n-k}} (\boldsymbol{u}, \boldsymbol{v}) \\ &= \left(\bigoplus_{\boldsymbol{v}\in\mathbb{F}_2^{n-k}} \boldsymbol{u}, \bigoplus_{\boldsymbol{v}\in\mathbb{F}_2^{n-k}} \boldsymbol{v}\right) = (\mathbf{0}_k, \mathbf{0}_{n-k}) = \mathbf{0} \end{aligned}$$

ya que cada una de las componentes de los vectores $\bigoplus_{\boldsymbol{v}\in\mathbb{F}_2^{n-k}} \boldsymbol{u}$ y $\bigoplus_{\boldsymbol{v}\in\mathbb{F}_2^{n-k}} \boldsymbol{v}$ es la suma de un número par de 1.

El resultado se sigue ahora del hecho de que si consideramos $\sigma \in S_n$ tal que $\sigma(j) = i_j$, para $j = 1, 2, \ldots, k$, entonces, por el teorema 2, los elementos de $\operatorname{Sop}(f)$ se obtienen a partir de los elementos de $\{\boldsymbol{u}\} \times \mathbb{F}_2^{n-k}$ permutando sus componentes de acuerdo con la permutación σ^{-1}. ∎

Notemos que la condición $1 \leq k \leq n-2$ del lema anterior es necesaria, ya que si $k = n$, entonces

$$\mathrm{Sop}\,(f) = \{\boldsymbol{u}\} \subseteq \mathbb{F}_2^n$$

mientras que si $k = n-1$, y $f(\boldsymbol{x})$ es el monomio de grado $n-1$ que no contiene la variable x_j, entonces, por el teorema 2,

$$\mathrm{Sop}\,(f) = \{\boldsymbol{u}, \boldsymbol{u}_j\} \quad \text{con} \quad \boldsymbol{u}_j = \boldsymbol{u} \oplus \boldsymbol{2^{n-j}}$$

y claramente $\boldsymbol{u} \oplus \boldsymbol{u}_j \neq \boldsymbol{0}$.

Teorema 3: *Si $f \in \mathcal{B}_n$ y $\mathrm{gr}\,(f) \leq n-2$, entonces*

$$\bigoplus_{\boldsymbol{a} \in \mathrm{Sop}(f)} \boldsymbol{a} = \boldsymbol{0}.$$

Demostración: Supongamos que

$$f(\boldsymbol{x}) = \bigoplus_{j=1}^{m} f_j(\boldsymbol{x})$$

con $f_j(\boldsymbol{x})$, para $j = 1, 2, \ldots, m$, un monomio de grado menor o igual que $n-2$.

Procederemos por inducción sobre m. Para $m = 1$ el resultado es cierto por el lema 1.

Supongamos que el resultado es cierto para $m-1$ y veamos que también lo es para m. Notemos en primer lugar que

$$\mathrm{Sop}\,(f) = \mathop{\Delta}_{j=1}^{m} \mathrm{Sop}\,(f_j) = \left(\mathop{\Delta}_{j=1}^{m-1} \mathrm{Sop}\,(f_j) \right) \Delta\, \mathrm{Sop}\,(f_m)\,.$$

Por comodidad en la notación, sean

$$A = \mathrm{Sop}\,(f)\,, \quad B = \mathop{\Delta}_{j=1}^{m-1} \mathrm{Sop}\,(f_j) \quad \text{y} \quad C = \mathrm{Sop}\,(f_m)\,.$$

De las propiedades de la unión, intersección y diferencia simétrica de conjuntos tenemos, por la hipótesis de inducción,

$$\boldsymbol{0} = \bigoplus_{\boldsymbol{b} \in B} \boldsymbol{b} = \bigoplus_{\boldsymbol{b} \in A \cap B} \boldsymbol{b} \oplus \bigoplus_{\boldsymbol{d} \in B \cap C} \boldsymbol{d}$$

y, por el lema 1,

$$\boldsymbol{0} = \bigoplus_{\boldsymbol{c} \in C} \boldsymbol{c} = \bigoplus_{\boldsymbol{c} \in A \cap C} \boldsymbol{c} \oplus \bigoplus_{\boldsymbol{e} \in B \cap C} \boldsymbol{e}.$$

Ahora, sumando las dos expresiones anteriores,

$$\boldsymbol{0} = \bigoplus_{\boldsymbol{b} \in A \cap B} \boldsymbol{b} \oplus \bigoplus_{\boldsymbol{c} \in A \cap C} \boldsymbol{c} = \bigoplus_{\boldsymbol{a} \in A} \boldsymbol{a}$$

ya que

$$\bigoplus_{\boldsymbol{d} \in B \cap C} \boldsymbol{d} \oplus \bigoplus_{\boldsymbol{e} \in B \cap C} \boldsymbol{e} = \boldsymbol{0}.$$

■

El recíproco del teorema anterior no es cierto en general. Sin embargo, antes de poder establecer bajo qué condiciones es cierto dicho recíproco, necesitamos introducir algunos resultados.

Otra consecuencia inmediata que se deduce del teorema 2 es que el grado de $f \in \mathcal{B}_n$ es n si y sólo si $w(f)$ es un número impar, tal como establecemos en el resultado siguiente.

Teorema 4: *Si $f \in \mathcal{B}_n$, entonces $\mathrm{gr}\,(f) = n$ si y sólo si $w(f)$ es un número impar.*

Demostración: Claramente $f = g \oplus h$, con $g, h \in \mathcal{B}_n$ tales que

$$\mathrm{gr}\,(g) \leq n-1 \quad \text{y} \quad h(\boldsymbol{x}) = a x_1 x_2 \cdots x_n \text{ con } a \in \mathbb{Z}_2.$$

Notemos que $w(h) = a$ y que $\mathrm{gr}\,(f) = n$ si y sólo si $a = 1$.

Además, si $g(\boldsymbol{x}) = \bigoplus_{j=1}^{m} g_j(\boldsymbol{x})$, con $g_j(\boldsymbol{x})$ un monomio tal que $\mathrm{gr}\,(g_j) \leq n-1$, para $j = 1, 2, \ldots, m$, entonces, por la expresión (1) y el teorema 2 tenemos que

$$w(f) \equiv \sum_{j=1}^{m} w(g_j) + w(h) \pmod 2 \equiv a \pmod 2$$

y así, $\mathrm{gr}\,(f) = n$ si y sólo si $w(f)$ es impar. ■

Este resultado permite probar, como habíamos anunciado antes, que el recíproco del teorema 3 no es cierto.

Ejemplo 1: Si $f \in \mathcal{B}_3$ y $\mathrm{Sop}\,(f) = \{\boldsymbol{3}, \boldsymbol{5}, \boldsymbol{6}\}$, por el teorema 4 tenemos que $\mathrm{gr}\,(f) = 3$ y sin embargo $\boldsymbol{3} \oplus \boldsymbol{5} \oplus \boldsymbol{6} = \boldsymbol{0}$.□

El resultado siguiente pone de manifiesto que la situación que describe el ejemplo anterior sólo se puede presentar cuando $|\mathrm{Sop}\,(f)|$ es impar, es decir, cuando $\mathrm{gr}\,(f) = n$.

Teorema 5: *Supongamos que $f \in \mathcal{B}_n$ tal que $|\mathrm{Sop}\,(f)|$ es un número par. Si*

$$\bigoplus_{\boldsymbol{a} \in \mathrm{Sop}(f)} \boldsymbol{a} = \boldsymbol{0}$$

entonces $1 \leq \mathrm{gr}\,(f) \leq n-2$.

Demostración: Como $|\mathrm{Sop}\,(f)|$ es par, por el teorema 4, tenemos que $\mathrm{gr}\,(f) \leq n-1$.

Si $\mathrm{gr}\,(f) = n-1$, entonces f tiene al menos un monomio de grado $n-1$. Supongamos que

$$f(\boldsymbol{x}) = \bigoplus_{j=1}^{m} g_j(\boldsymbol{x}) \oplus h(\boldsymbol{x})$$

con $g_j(\boldsymbol{x})$, para $j = 1, 2, \ldots, m$, un monomio de grado $n-1$ que no contiene la variable x_{i_j} y que $\mathrm{gr}\,(h) \leq n-2$.

Procediendo como en la demostración del teorema 3, teniendo en cuenta que por el teorema 3, $\bigoplus_{\boldsymbol{a} \in \mathrm{Sop}(h)} \boldsymbol{a} = \boldsymbol{0}$, y de acuerdo con el comentario que precede al teorema 3, tenemos que

$$\boldsymbol{0} = \bigoplus_{\boldsymbol{a} \in \mathrm{Sop}(f)} \boldsymbol{a} = \bigoplus_{j=1}^{m} \bigoplus_{\boldsymbol{a} \in \mathrm{Sop}(g_j)} \boldsymbol{a} \oplus \bigoplus_{\boldsymbol{a} \in \mathrm{Sop}(h)} \boldsymbol{a}$$

$$= \begin{cases} \displaystyle\bigoplus_{j=1}^{m} \boldsymbol{2^{n-i_j}}, & \text{si } m \text{ es par,} \\ \boldsymbol{u} \oplus \displaystyle\bigoplus_{j=1}^{m} \boldsymbol{2^{n-i_j}}, & \text{si } m \text{ es impar} \end{cases}$$

que es una contradicción. Por tanto, $\mathrm{gr}\,(f) \leq n-2$. ■

Así, como consecuencia inmediata de los teoremas 3 y 5 tenemos el siguiente resultado.

Corolario 1: *Sea $f \in \mathcal{B}_n$ tal que $|\mathrm{Sop}\,(f)|$ es un número par. Entonces $\mathrm{gr}\,(f) \leq n-2$ si y sólo si*

$$\bigoplus_{\boldsymbol{a}\in \mathrm{Sop}(f)} \boldsymbol{a} = \mathbf{0}.$$

Antes de continuar, veamos cómo podemos utilizar los resultados vistos hasta ahora para obtener el grado de una función booleana a partir de su soporte.

Ejemplo 2: Sea $f \in \mathcal{B}_4$ tal que

$$\mathrm{Sop}\,(f) = \{\mathbf{0}, \mathbf{5}, \mathbf{6}, \mathbf{7}, \mathbf{9}, \mathbf{10}, \mathbf{11}, \mathbf{13}, \mathbf{14}, \mathbf{15}\}.$$

Puesto que $|\mathrm{Sop}\,(f)| = 10$ es un número par y

$$\mathbf{0} \oplus \mathbf{5} \oplus \mathbf{6} \oplus \mathbf{7} \oplus \mathbf{9} \oplus \mathbf{10} \oplus \mathbf{11} \oplus \mathbf{13} \oplus \mathbf{14} \oplus \mathbf{15} = \mathbf{0}$$

por el corolario 1 tenemos que $\mathrm{gr}\,(f) \leq 4-2 = 2$.

Por otro lado, como $w(f) = |\mathrm{Sop}\,(f)| \neq 8$, tenemos que $f(\boldsymbol{x})$ no es equilibrada y, por tanto, $f(\boldsymbol{x})$ no es lineal ni afín. Como tampoco es constante, necesariamente, $\mathrm{gr}\,(f) > 1$.

En consecuencia, $\mathrm{gr}\,(f) = 2$. □

Como acabamos de ver, es posible determinar el grado de una función booleana $f \in \mathcal{B}_n$ a partir de su soporte sin necesidad de conocer su FNA. Veremos seguidamente que, es posible también determinar, de forma sencilla, si un determinado monomio forma parte de la FNA de f.

Sea $f \in \mathcal{B}_n$ y supongamos que conocemos $\mathrm{Sop}\,(f)$. Puesto que $w(f) = |\mathrm{Sop}\,(f)|$ el teorema 4 permite afirmar que en la FNA de $f(\boldsymbol{x})$ está el monomio $x_1 x_2 \cdots x_n$ si y sólo si $|\mathrm{Sop}\,(f)|$ es un número impar. El teorema siguiente establece una condición necesaria y suficiente para que la FNA de $f(\boldsymbol{x})$ contenga cualquier monomio de grado k con $1 \leq k < n$. Antes, sin embargo, necesitamos el resultado siguiente que facilitará la demostración de dicho teorema.

Lema 2: *Supongamos que $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ y consideremos la aplicación $\varphi : \mathbb{F}_2^k \to \mathbb{F}_2^n$ dada por $\varphi(y_1, y_2, \ldots, y_k) = (x_1, x_2, \ldots, x_n)$ con*

$$x_l = \begin{cases} 0, & \text{si } l \notin \{i_1, i_2, \ldots, i_k\}, \\ y_j, & \text{si } l = i_j, \text{ para } j = 1, 2, \ldots, k. \end{cases}$$

Si $f \in \mathcal{B}_n$ y consideramos $h \in \mathcal{B}_k$ tal que

$$h(y_1, y_2, \ldots, y_k) = f(\varphi(y_1, y_2, \ldots, y_k)),$$

entonces

$$|\mathrm{Sop}\,(h)| = \left|\mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{n-i_1}}, \mathbf{2^{n-i_2}}, \ldots, \mathbf{2^{n-i_k}}\right\}\right|.$$

Demostración: Notemos en primer lugar que φ es un monomorfismo de espacios vectoriales y que $\mathrm{Im}(\varphi) = \mathrm{Env}\left\{\mathbf{2^{n-i_1}}, \mathbf{2^{n-i_2}}, \ldots, \mathbf{2^{n-i_k}}\right\}$.

Si $\boldsymbol{u} \in \mathrm{Sop}\,(h)$, entonces $1 = h(\boldsymbol{u}) = f(\varphi(\boldsymbol{u}))$ con lo que $\varphi(\boldsymbol{u}) \in \mathrm{Sop}\,(f)$ y como claramente $\varphi(\boldsymbol{u}) \in \mathrm{Im}(\varphi)$, tenemos que $\varphi(\boldsymbol{u}) \in \mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{n-i_1}}, \mathbf{2^{n-i_2}}, \ldots, \mathbf{2^{n-i_k}}\right\}$, por tanto,

$$\varphi(\mathrm{Sop}\,(h)) \subseteq \mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{n-i_1}}, \mathbf{2^{n-i_2}}, \ldots, \mathbf{2^{n-i_k}}\right\}.$$

Supongamos ahora que

$$\boldsymbol{v} = (v_1, v_2, \ldots, v_n) \in \mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{n-i_1}}, \mathbf{2^{n-i_2}}, \ldots, \mathbf{2^{n-i_k}}\right\}$$

entonces $f(\boldsymbol{v}) = 1$ y $v_l = 0$ si $l \notin \{i_1, i_2, \ldots, i_k\}$. Claramente

$$\boldsymbol{u} = (v_1, v_2, \ldots, v_k) \in \mathbb{F}_2^k \quad \text{y} \quad \varphi(\boldsymbol{u}) = \boldsymbol{v},$$

por tanto $h(\boldsymbol{u}) = f(\varphi(\boldsymbol{u})) = f(\boldsymbol{v}) = 1$, con lo que $\boldsymbol{u} \in \mathrm{Sop}\,(h)$ y así $\boldsymbol{v} = \varphi(\boldsymbol{u}) \in \varphi(\mathrm{Sop}\,(h))$. En consecuencia

$$\mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{n-i_1}}, \mathbf{2^{n-i_2}}, \ldots, \mathbf{2^{n-i_k}}\right\} \subseteq \varphi(\mathrm{Sop}\,(h)).$$

De esta inclusión y de la anterior tenemos que

$$\varphi(\mathrm{Sop}\,(h)) = \mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{n-i_1}}, \mathbf{2^{n-i_2}}, \ldots, \mathbf{2^{n-i_k}}\right\}.$$

El resultado se sigue ahora del hecho de que $|\varphi(\mathrm{Sop}\,(h))| = |\mathrm{Sop}\,(h)|$ por ser φ inyectiva. ■

Teorema 6: *Supongamos que conocemos $\mathrm{Sop}\,(f)$ para $f \in \mathcal{B}_n$. La FNA de $f(\boldsymbol{x})$ contiene el monomio $x_{i_1} x_{i_2} \cdots x_{i_k}$ con $1 \leq k < n$ si y sólo si*

$$\left|\mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{n-i_1}}, \mathbf{2^{n-i_2}}, \ldots, \mathbf{2^{n-i_k}}\right\}\right|$$

es un número impar.

Demostración: Con la notación del lema 2 tenemos que la FNA de $f(\boldsymbol{x})$ contiene el monomio $x_{i_1} x_{i_2} \cdots x_{i_k}$ si y sólo si la FNA de $h(\boldsymbol{y})$ contiene el monomio $y_1 y_2 \cdots y_k$. Ahora bien, por el teorema 4, la FNA de $h(\boldsymbol{y})$ contiene el monomio $y_1 y_2 \cdots y_k$ si y sólo si $|\mathrm{Sop}\,(h)|$ es un número impar. Finalmente, por el lema 2, la FNA de $f(\boldsymbol{x})$ contiene el monomio $x_{i_1} x_{i_2} \cdots x_{i_k}$ si y sólo si $\left|\mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{n-i_1}}, \mathbf{2^{n-i_2}}, \ldots, \mathbf{2^{n-i_k}}\right\}\right|$ es un número impar. ■

Notemos que el teorema anterior permite afirmar que $\mathrm{gr}\,(f) \geq k$ si y sólo si existen k números enteros tales que $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ y

$$\left|\mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{n-i_1}}, \mathbf{2^{n-i_2}}, \ldots, \mathbf{2^{n-i_k}}\right\}\right|$$

es un número impar, o equivalentemente,

$$\mu_f(\boldsymbol{u}) = \left|\mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{n-i_1}}, \mathbf{2^{n-i_2}}, \ldots, \mathbf{2^{n-i_k}}\right\}\right| \bmod 2$$

para $\boldsymbol{u} = (u_1, u_2, \ldots, u_n) \in \mathbb{F}_2^n$ con

$$u_l = \begin{cases} 0, & \text{si } l \notin \{i_1, i_2, \ldots, i_k\}, \\ 1, & \text{si } l = i_j, \text{ para } j = 1, 2, \ldots, k. \end{cases}$$

Aplicando sucesivamente el resultado anterior, podemos determinar tanto el grado como la FNA de una función booleana de la que conocemos su soporte. El ejemplo siguiente nos ayudará a entender este proceso.

Ejemplo 3: Consideremos de nuevo la función $f \in \mathcal{B}_4$ del ejemplo 2. Hemos visto en dicho ejemplo que $\mathrm{gr}\,(f) = 2$. Si queremos obtener la FNA de $f(\boldsymbol{x})$, necesitamos saber cuáles son los monomios de grado 2 y de grado 1 que contiene.

Los posibles monomios de grado 2 son

$$x_1 x_2, \quad x_1 x_3, \quad x_1 x_4, \quad x_2 x_3, \quad x_2 x_4 \quad \text{y} \quad x_3 x_4.$$

Que alguno de dichos monomios esté en la FNA de $f(\boldsymbol{x})$ depende, de acuerdo con el teorema 6, de que alguno de los cardinales de los conjuntos

$$\begin{aligned}
S_1 &= \mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{4-1}}, \mathbf{2^{4-2}}\right\},\\
S_2 &= \mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{4-1}}, \mathbf{2^{4-3}}\right\},\\
S_3 &= \mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{4-1}}, \mathbf{2^{4-4}}\right\},\\
S_4 &= \mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{4-2}}, \mathbf{2^{4-3}}\right\},\\
S_5 &= \mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{4-2}}, \mathbf{2^{4-4}}\right\},\\
S_6 &= \mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{4-3}}, \mathbf{2^{4-4}}\right\},
\end{aligned}$$

sea impar. Puesto que

$$\begin{aligned}
&S_1 = \{\mathbf{0}\}, \quad S_2 = \{\mathbf{0}, \mathbf{10}\}, \quad S_3 = \{\mathbf{0}, \mathbf{9}\},\\
&S_4 = \{\mathbf{0}, \mathbf{6}\}, \quad S_5 = \{\mathbf{0}, \mathbf{5}\} \quad \text{y} \quad S_6 = \{\mathbf{0}\},
\end{aligned}$$

podemos afirmar que la FNA de $f(\boldsymbol{x})$ contiene los monomios x_1x_2 y x_3x_4.

Para determinar los monomios de grado 1 de la FNA de $\boldsymbol{x}$ procedemos de la misma forma. Puesto que

$$\begin{aligned}
&\mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{4-1}}\right\} = \{\mathbf{0}\},\ \mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{4-2}}\right\} = \{\mathbf{0}\},\\
&\mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{4-3}}\right\} = \{\mathbf{0}\},\ \mathrm{Sop}\,(f) \cap \mathrm{Env}\left\{\mathbf{2^{4-4}}\right\} = \{\mathbf{0}\},
\end{aligned}$$

tenemos, de acuerdo con el teorema 6, que los monomios x_1, x_2, x_3 y x_4 forman parte de la FNA de $f(\boldsymbol{x})$.

Finalmente, como $\mathbf{0} \in \mathrm{Sop}\,(f)$, tenemos que el término constante 1, forma parte de la FNA de $f(\boldsymbol{x})$.

En consecuencia, la FNA de $f(\boldsymbol{x})$ es

$$f(\boldsymbol{x}) = 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_1x_2 \oplus x_3x_4. \qquad \square$$

Como acabamos de ver en el ejemplo anterior, para determinar completamente la FNA de $f(\boldsymbol{x})$ a partir de $\mathrm{Sop}\,(f)$ necesitamos calcular el cardinal de una serie de conjuntos obtenidos a partir de $\mathrm{Sop}\,(f)$, por lo que el procedimiento, en general, no constituye de por sí un algoritmo eficiente.

Sin embargo, si $1 \leq k < n$, mediante un razonamiento análogo al seguido para obtener la expresión (3), podemos separar las n variables en dos conjuntos formados por k y $n-k$ variables, respectivamente, tal como describimos en el resultado siguiente. La demostración es fácil pero larga y, por tanto, la omitimos.

Teorema 7: *Supongamos que $f \in \mathcal{B}_n$. Si $1 \leq k < n$ entonces*

$$f(\boldsymbol{y}, \boldsymbol{x}) = \bigoplus_{\boldsymbol{b} \in \mathbb{F}_2^k} \left(\bigoplus_{\boldsymbol{a} \in E(\boldsymbol{b})} f_{\boldsymbol{a}}(\boldsymbol{x}) \right) \boldsymbol{y}^{\boldsymbol{b}}$$

donde $f_{\boldsymbol{a}} \in \mathcal{B}_k$, para $\boldsymbol{a} \in \mathbb{F}_2^k$, satisface $f_{\boldsymbol{a}}(\boldsymbol{x}) = f(\boldsymbol{a}, \boldsymbol{x})$. Además,

$$\mathrm{gr}\,(f) = \max_{\boldsymbol{b} \in \mathbb{F}_2^k} \left\{ \mathrm{gr} \left(\bigoplus_{\boldsymbol{a} \in E(\boldsymbol{b})} f_{\boldsymbol{a}} \right) + w(\boldsymbol{b}) \right\}.$$

El resultado siguiente, cuya demostración es inmediata, establece la relación entre $\mathrm{Sop}\,(f)$ y $\mathrm{Sop}\,(f_{\boldsymbol{a}})$, para todo $\boldsymbol{a} \in \mathbb{F}_2^k$.

Teorema 8: *Con la notación del teorema 7,*

$$\mathrm{Sop}\,(f_{\boldsymbol{a}}) = \{\boldsymbol{v} \in \mathbb{F}_2^{n-k} \mid (\boldsymbol{a}, \boldsymbol{v}) \in \mathrm{Sop}\,(f)\}$$

y

$$\mathrm{Sop} \left(\bigoplus_{\boldsymbol{a} \in E(\boldsymbol{b})} f_{\boldsymbol{a}} \right) = \bigtriangleup_{\boldsymbol{a} \in E(\boldsymbol{b})} \mathrm{Sop}\,(f_{\boldsymbol{a}})\,.$$

Antes de continuar, veamos un ejemplo que nos ayudará a entender el proceso a seguir.

Ejemplo 4: Sea $f \in \mathcal{B}_5$ tal que

$$\begin{aligned}
&\mathrm{Sop}\,(f)\\
&= \{\mathbf{6}, \mathbf{7}, \mathbf{12}, \mathbf{13}, \mathbf{16}, \mathbf{17}, \mathbf{18}, \mathbf{20}, \mathbf{21}, \mathbf{23}, \mathbf{24}, \mathbf{25}, \mathbf{26}, \mathbf{30}\}.
\end{aligned}$$

Puesto que $|\mathrm{Sop}\,(f)|$ es par y

$$\begin{aligned}
&\mathbf{6} \oplus \mathbf{7} \oplus \mathbf{12} \oplus \mathbf{13} \oplus \mathbf{16} \oplus \mathbf{17} \oplus \mathbf{18}\\
&\qquad \oplus \mathbf{20} \oplus \mathbf{21} \oplus \mathbf{23} \oplus \mathbf{24} \oplus \mathbf{25} \oplus \mathbf{26} \oplus \mathbf{30} = \mathbf{0}
\end{aligned}$$

por el corolario 1, tenemos que $\mathrm{gr}\,(f) \leq 5 - 2 = 3$.

Ahora, de acuerdo con el teorema 7, tenemos que

$$\mathrm{gr}\,(f) = \max\{\mathrm{gr}\,(f_0)\,, \mathrm{gr}\,(f_0 \oplus f_1) + 1\}$$

con $f_0, f_1 \in \mathcal{B}_4$ tales que

$$\begin{aligned}
f_0(x_2, x_3, x_4, x_5) &= f(0, x_2, x_3, x_4, x_5),\\
f_1(x_2, x_3, x_4, x_5) &= f(1, x_2, x_3, x_4, x_5),
\end{aligned}$$

y por el teorema 8,

$$\begin{aligned}
\mathrm{Sop}\,(f_0) &= \{\mathbf{6}, \mathbf{7}, \mathbf{12}, \mathbf{13}\}\\
\mathrm{Sop}\,(f_1) &= \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{4}, \mathbf{5}, \mathbf{7}, \mathbf{8}, \mathbf{9}, \mathbf{10}, \mathbf{14}\}.
\end{aligned}$$

Por tanto

$$\begin{aligned}
\mathrm{Sop}\,(f_0 \oplus f_1) &= \mathrm{Sop}\,(f_0) \,\triangle\, \mathrm{Sop}\,(f_1)\\
&= \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{8}, \mathbf{9}, \mathbf{10}, \mathbf{12}, \mathbf{13}, \mathbf{14}\}.
\end{aligned}$$

Además, como $\mathbf{6} \oplus \mathbf{7} \oplus \mathbf{12} \oplus \mathbf{13} = \mathbf{0}$ y

$$\begin{aligned}
&\mathbf{0} \oplus \mathbf{1} \oplus \mathbf{2} \oplus \mathbf{4} \oplus \mathbf{5} \oplus \mathbf{6} \oplus \mathbf{8}\\
&\qquad \oplus \mathbf{9} \oplus \mathbf{10} \oplus \mathbf{12} \oplus \mathbf{13} \oplus \mathbf{14} = \mathbf{0}
\end{aligned}$$

por el corolario 1 tenemos que

$$\mathrm{gr}\,(f_0) \leq 4 - 2 = 2 \quad \text{y} \quad \mathrm{gr}\,(f_0 \oplus f_1) \leq 4 - 2 = 2$$

y así $\mathrm{gr}\,(f) \leq \max\{2, 2+1\} = 3$.

Ahora, de nuevo por el teorema 7 tenemos que

$$\begin{aligned}
\mathrm{gr}\,(f) = \max\{&\mathrm{gr}\,(f_{\mathbf{0}})\,, \mathrm{gr}\,(f_{\mathbf{0}} \oplus f_{\mathbf{1}}) + 1,\\
&\mathrm{gr}\,(f_{\mathbf{0}} \oplus f_{\mathbf{2}}) + 1, \mathrm{gr}\,(f_{\mathbf{0}} \oplus f_{\mathbf{1}} \oplus f_{\mathbf{2}} \oplus f_{\mathbf{3}}) + 2\}
\end{aligned}$$

con $f_{\mathbf{0}}, f_{\mathbf{1}}, f_{\mathbf{2}}, f_{\mathbf{3}} \in \mathcal{B}_3$ tales que

$$\begin{aligned}
f_{\mathbf{0}}(x_3, x_4, x_5) &= f(0, 0, x_3, x_4, x_5),\\
f_{\mathbf{1}}(x_3, x_4, x_5) &= f(0, 1, x_3, x_4, x_5),\\
f_{\mathbf{2}}(x_3, x_4, x_5) &= f(1, 0, x_3, x_4, x_5),\\
f_{\mathbf{3}}(x_3, x_4, x_5) &= f(1, 1, x_3, x_4, x_5)
\end{aligned}$$

y, por el teorema 8,

$$\mathrm{Sop}\,(f_{\mathbf{0}}) = \{\mathbf{6},\mathbf{7}\}, \quad \mathrm{Sop}\,(f_{\mathbf{1}}) = \{\mathbf{4},\mathbf{5}\}$$
$$\mathrm{Sop}\,(f_{\mathbf{2}}) = \{\mathbf{0},\mathbf{1},\mathbf{2},\mathbf{4},\mathbf{5},\mathbf{7}\} \quad \text{y} \quad \mathrm{Sop}\,(f_{\mathbf{3}}) = \{\mathbf{0},\mathbf{1},\mathbf{2},\mathbf{6}\}$$

con lo que

$$\mathrm{Sop}\,(f_{\mathbf{0}} \oplus f_{\mathbf{1}}) = \mathrm{Sop}\,(f_{\mathbf{0}}) \,\Delta\, \mathrm{Sop}\,(f_{\mathbf{1}}) = \{\mathbf{4},\mathbf{5},\mathbf{6},\mathbf{7}\}$$
$$\mathrm{Sop}\,(f_{\mathbf{0}} \oplus f_{\mathbf{2}}) = \mathrm{Sop}\,(f_{\mathbf{0}}) \,\Delta\, \mathrm{Sop}\,(f_{\mathbf{2}}) = \{\mathbf{0},\mathbf{1},\mathbf{2},\mathbf{4},\mathbf{5},\mathbf{6}\}$$

y

$$\begin{aligned}&\mathrm{Sop}\,(f_{\mathbf{0}} \oplus f_{\mathbf{1}} \oplus f_{\mathbf{2}} \oplus f_{\mathbf{3}})\\ &= \mathrm{Sop}\,(f_{\mathbf{0}}) \,\Delta\, \mathrm{Sop}\,(f_{\mathbf{1}}) \,\Delta\, \mathrm{Sop}\,(f_{\mathbf{2}}) \,\Delta\, \mathrm{Sop}\,(f_{\mathbf{3}}) = \emptyset.\end{aligned}$$

Además, como $\mathbf{6} \oplus \mathbf{7} = \mathbf{1} \neq \mathbf{0}$, $\mathbf{4} \oplus \mathbf{5} \oplus \mathbf{6} \oplus \mathbf{7} = \mathbf{0}$ y

$$\mathbf{0} \oplus \mathbf{1} \oplus \mathbf{2} \oplus \mathbf{4} \oplus \mathbf{5} \oplus \mathbf{6} = \mathbf{4} \neq \mathbf{0}$$

por el corolario 1 tenemos que

$$\mathrm{gr}\,(f_{\mathbf{0}}) = 2, \quad \mathrm{gr}\,(f_{\mathbf{0}} \oplus f_{\mathbf{1}}) \leq 1, \quad \mathrm{gr}\,(f_{\mathbf{0}} \oplus f_{\mathbf{2}}) = 2$$

y $f_{\mathbf{0}} \oplus f_{\mathbf{1}} \oplus f_{\mathbf{2}} \oplus f_{\mathbf{3}} = 0$, con lo que $\mathrm{gr}\,(f) = 3$. □

Los resultados anteriores permiten establecer el algoritmo siguiente para determinar el grado de cualquier $f \in \mathcal{B}_n$ a partir de $\mathrm{Sop}\,(f)$.

Algoritmo: Obtención de $\mathrm{gr}\,(f)$, para $f \in \mathcal{B}_n$, a partir de $\mathrm{Sop}\,(f)$.

1) Si $|\mathrm{Sop}\,(f)|$ es impar, entonces $\mathrm{gr}\,(f) = n$.
2) En otro caso, sea $\boldsymbol{s} = \bigoplus_{\boldsymbol{a} \in \mathrm{Sop}(f)} \boldsymbol{a}$.
3) Si $\boldsymbol{s} \neq \mathbf{0}$, entonces $\mathrm{gr}\,(f) = n - 1$.
4) En otro caso, sea $\mathrm{maxgr}\,(f) = n - 2$ el máximo valor que puede tomar $\mathrm{gr}\,(f)$ y suponer que $k = 1$.
   a) Para $\boldsymbol{b} \in \mathbb{F}_2^k$:
      i) Obtener $g_{\boldsymbol{b}} = \bigoplus_{\boldsymbol{a} \in E(\boldsymbol{b})} f_{\boldsymbol{a}}$.
      ii) Si $|\mathrm{Sop}\,(g_{\boldsymbol{b}})|$ es impar, entonces $\mathrm{gr}\,(g_{\boldsymbol{b}}) = n - k$.
      iii) En otro caso, sea $\boldsymbol{s}_{\boldsymbol{b}} = \bigoplus_{\boldsymbol{a} \in \mathrm{Sop}(g_{\boldsymbol{b}})} \boldsymbol{a}$.
      iv) Si $\boldsymbol{s}_{\boldsymbol{b}} \neq \mathbf{0}$, entonces $\mathrm{gr}\,(g_{\boldsymbol{b}}) = n - k - 1$. En otro caso, $\mathrm{gr}\,(g_{\boldsymbol{b}}) \leq n - k - 2$.
   b) Si $\mathrm{maxgr}\,(f) = \max_{\boldsymbol{b} \in \mathbb{F}_2^k} \{\mathrm{gr}\,(g_{\boldsymbol{b}}) + w(\boldsymbol{b})\}$, el proceso termina y $\mathrm{gr}\,(f) = \mathrm{maxgr}\,(f)$. En otro caso, hacer $\mathrm{maxgr}\,(f) = \max_{\boldsymbol{b} \in \mathbb{F}_2^k} \{\mathrm{gr}\,(g_{\boldsymbol{b}}) + w(\boldsymbol{b})\}$, aumentar k en una unidad e ir al paso 4a).

IV. Conclusiones y lineas futuras

En este artículo presentamos una serie de propiedades de una función booleana que relacionan el soporte y la FNA de la misma. Estas propiedades nos han permitido establecer un algoritmo para determinar el grado de una función booleana cuando conocemos el soporte de la misma (y obviamente no conocemos su FNA). En un trabajo futuro abordaremos los aspectos computacionales del algoritmo propuesto.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por los proyectos MTM2008-06674-C02-01 del Ministerio de Ciencia e Innovación del Gobierno de España y ACOMP/2010/039 de la Generalitat Valenciana. La investigación de Verónica Requena también ha sido financiada con una ayuda del Vicerrectorado de Investigación, Desarrollo e Innovación de la Universitat d'Alacant destinada a la formación de doctores.

Referencias

[1] C. M. Adams, "Constructing symmetric ciphers using the CAST design procedure," *Designs, Codes and Cryptography*, vol. 12, pp. 283–316, 1997.

[2] A. Braeken, V. Nikov, S. Nikova, and B. Preneel, "On Boolean functions with generalized cryptographic properties," in *Progress in Cryptology – INDOCRYPT 2004*, ser. Lecture Notes in Computer Science, A. Canteaut and K. Viswanathan, Eds. Berlin: Springer-Verlag, 2004, vol. 3348, pp. 120–135.

[3] Y. Borissov, A. Braeken, S. Nikova, and B. Preneel, "On the covering radii of binary Reed-Muller codes in the set of resilient Boolean functions," *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 1182–1189, 2005.

[4] C. Carlet and Y. Tarannikov, "Covering sequences of Boolean functions and their cryptographic significance," *Designs, Codes and Cryptography*, vol. 25, pp. 263–279, 2002.

[5] K. C. Gupta and P. Sarkar, "Improved construction of nonlinear resilient S-boxes," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 339–348, 2005.

[6] K. Kurosawa, T. Iwata, and T. Yoshiwara, "New covering radius of Reed-Muller codes for t-resilient functions," *IEEE Transactions on Information Theory*, vol. 50, no. 3, pp. 468–475, 2004.

[7] K. Kurosawa and R. Matsumoto, "Almost security of cryptographic Boolean functions," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2752–2761, 2004.

[8] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Advances in Cryptology – EUROCRYPT '93*, ser. Lecture Notes in Computer Science, T. Helleseth, Ed. Berlin: Springer-Verlag, 1994, vol. 765, pp. 386–397.

[9] W. Meier and O. Staffelbach, "Nonlinearity criteria for cryptographic functions," in *Advances in Cryptology – EUROCRYPT'89*, ser. Lecture Notes in Computer Science, J. Quisquater and J. Vandewalle, Eds. Berlin: Springer-Verlag, 1990, vol. 434, pp. 549–562.

[10] K. Nyberg, "Perfect nonlinear S-boxes," in *Advances in Cryptology – EUROCRYPT '91*, ser. Lecture Notes in Computer Science, D. W. Davies, Ed. Berlin: Springer-Verlag, 1991, vol. 547, pp. 378–386.

[11] D. Olejár and M. Stanek, "On cryptographic properties of random Boolean functions," *Journal of Universal Computer Science*, vol. 4, no. 8, pp. 705–717, 1998.

[12] E. Pasalic and T. Johansson, "Further results on the relation between nonlinearity and resiliency for Boolean functions," in *Crytography and Coding*, ser. Lecture Notes in Computer Science, M. Walker, Ed. Berlin: Springer-Verlag, 1999, vol. 1746, pp. 35–44.

[13] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, and J. Vandewalle, "Propagation characteristics of Boolean functions," in *Advances in Cryptology – EUROCRYPT'90*, ser. Lecture Notes in Computer Science, I. B. Damgard, Ed. Berlin: Springer-Verlag, 1991, vol. 473, pp. 161–173.

[14] C. Qu, J. Seberry, and J. Pieprzyk, "On the symmetric property of homogeneous Boolean functions," in *Proceedings of the Australasian Conference on Information Security and Privacy – ACISP'99*, ser. Lecture Notes in Computer Science, J. Pieprzyk, R. Safavi-Naini, and J. Seberry, Eds. Berlin: Springer-Verlag, 1999, vol. 1587, pp. 26–35.

[15] R. A. Rueppel, *Analysis and Design of Stream Ciphers*. New York, NY: Springer Verlag, 1986.

[16] R. A. Rueppel and O. J. Staffelbach, "Products of linear recurring sequences with maximum complexity," *IEEE Transactions on Information Theory*, vol. 33, no. 1, pp. 124–131, 1987.

Construcción de funciones bent de n variables a partir de una base de $\mathbb{F}_2^n$

Joan-Josep Climent
Departament d'Estadística i Investigació Operativa. Universitat d'Alacant
Email: jcliment@ua.es

Francisco J. García
Departament de Fonaments de l'Anàlisi Econòmica. Universitat d'Alacant
Email: francisco.garcia@ua.es

Verónica Requena
Departament d'Estadística i Investigació Operativa. Universitat d'Alacant
Email: jcliment@ua.es

***Resumen*—Utilizando una base de $\mathbb{F}_2^n$ (con n un número par) y la matriz asociada de un polinomio primitivo de grado $n/2$ y coeficientes en $\mathbb{F}_2$, construimos los soportes de algunas funciones bent de n variables.**

I. Introducción

Las funciones booleanas se utilizan en distintas aplicaciones criptográficas tales como cifrado en bloque, cifrado en flujo y funciones hash [5], [10], [18], [23]. Por ejemplo, la implementación de una caja de sustitución o S-box necesita funciones booleanas no lineales para resistir ataques tales como el criptoanálisis lineal o el criptoanálisis diferencial [3], [16], [21], [25].

Para un número par de variables, las funciones booleanas de máxima no linealidad son las llamadas funciones *bent* [11], [29], [31]. El nombre *bent* para dichas funciones se debe a Rothaus [28], aunque su origen se remonta a un artículo de McFarland [22] sobre conjuntos de diferencias en grupos no cíclicos; posteriormente Dillon [12] sistematizó y extendió las ideas de McFarland proporcionando una gran cantidad de propiedades. Desde entonces estas funciones han sido objeto de un intenso estudio como se desprende de la abundante literatura al respecto (véase por ejemplo [1], [2], [4], [6], [8], [9], [14], [15], [17], [19], [24], [26] y las referencias en ellas incluidas).

En estos momentos, se desconoce la existencia de un método que permita obtener todas las funciones bent y solamente se conoce su número para algunos casos particulares ($n = 2, 4, 6, 8$ con n el número de variables); la clasificación, y por tanto el número de tales funciones, para $n > 8$ continúa siendo un problema abierto.

Hay muchos métodos para construir funciones bent [7], [12], [13], [17], [22], [25], [28] los cuales han dado lugar a diferentes clases de funciones: Maiorana-McFarland, Rothauss, Partial Spread ($\mathcal{PS}$), etc. En este artículo nos centramos en la construcción de funciones bent que pertenecen a la clase $\mathcal{PS}$.

El resto del artículo está organizado de la siguiente manera: En la sección II, introducimos algunas definiciones básicas y la notación utilizada. En la sección III, presentamos un método general para caracterizar la construcción de funciones bent de n variables (con n un número par) de la clase $\mathcal{PS}$ utilizando una base de $\mathbb{F}_2^n$ y un polinomio primitivo de grado $n/2$ en $\mathbb{F}_2[\mathsf{X}]$. En la sección IV, introducimos un procedimiento práctico para la obtención del soporte de una función bent del tipo $\mathcal{PS}$, mostrando algunos ejemplos; además establecemos el número de dichos soportes que podemos construir con una base y un polinomio primitivo fijo; dicho número constituye, en definitiva, una cota inferior del número de funciones bent de n variables. Finalmente, en la sección V, presentamos algunos problemas abiertos relacionados con la construcción introducida en este artículo.

II. Preliminares

Sea $\mathbb{F}_2$ el cuerpo binario con la adición denotada por $\oplus$ y la multiplicación denotada por yuxtaposición. Para cualquier entero positivo n, es bien conocido que $\mathbb{F}_2^n$ es un espacio vectorial sobre $\mathbb{F}_2$ con la adición usual (denotada también por $\oplus$). Para una matriz cualquiera A de tamaño $n \times k$ con elementos en $\mathbb{F}_2$, denotamos por $\mathrm{col}\,(A)$ el espacio columna de A; por tanto, $\mathrm{col}\,(A)$ es un subespacio vectorial de $\mathbb{F}_2^n$. En este artículo, los vectores de $\mathbb{F}_2^n$ son vectores columna. Recordemos que la matriz de control de paridad H del $[2^k - 1, k]$-código binario de Hamming es la matriz cuya i-ésima columna de k dígitos del entero i para $1 \leq i \leq 2^k - 1$ (véase [27]); por tanto, es evidente que los vectores de $\mathrm{col}\,(A)$ son las columnas de la matriz AH.

Una **función booleana** de n variables es una aplicación $f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$. El conjunto de todas las funciones booleanas de n variables se denota por $\mathcal{B}_n$ y es un espacio vectorial con la adición (denotada también por $\oplus$) de dimensión 2^n sobre $\mathbb{F}_2$. Para $f \in \mathcal{B}_n$, llamamos **soporte** de f al conjunto

$$\mathrm{Sop}\,(f) = \{\boldsymbol{a} \in \mathbb{F}_2^n \mid f(\boldsymbol{a}) = 1\}.$$

El **peso de Hamming** de $f \in \mathcal{B}_n$, que denotamos por $w(f)$, es el cardinal de $\mathrm{Sop}\,(f)$.

Decimos que $f \in \mathcal{B}_n$ es una **función afín** si

$$f(\boldsymbol{x}) = \langle \boldsymbol{a}, \boldsymbol{x} \rangle \oplus b,$$

donde $\boldsymbol{a} \in \mathbb{F}_2^n$, $b \in \mathbb{F}_2$ y $\langle \boldsymbol{a}, \boldsymbol{x} \rangle$ es el producto escalar usual de los vectores $\boldsymbol{a}$ y $\boldsymbol{x}$. Si $b = 0$, decimos que f es una **función lineal**.

Definimos la **no linealidad** de una función $f \in \mathcal{B}_n$ como

$$\mathrm{NL}(f) = \min\{d(f, \varphi) \mid \varphi \in \mathcal{A}_n\}$$

donde $\mathcal{A}_n$ es el conjunto de todas las funciones afines y la distancia $d(f, g)$ entre dos funciones booleanas $f, g \in \mathcal{B}_n$ se

define como $d(f,g) = w(f \oplus g)$. La no linealidad de f está acotada superiormente (véase [31]) por

$$\mathrm{NL}(f) \leq 2^{n-1} - 2^{\frac{n}{2}-1}.$$

Llamamos **funciones bent** a las funciones booleanas que alcanzan la máxima no linealidad (véase [31]). Por tanto, las funciones bent solamente existen para n par.

Finalmente, es bien conocido (véase por ejemplo [30]) que si $f(\boldsymbol{x})$ es una función bent, entonces su función complementaria, $1 \oplus f(\boldsymbol{x})$, es también una función bent.

A partir de ahora suponemos que $n = 2k$ y nos centramos en la obtención de funciones bent de la clase $\mathcal{PS}$. El teorema siguiente, al que haremos referencia en diversas ocasiones, introduce dicha clase de funciones bent (véase [12]).

Teorema 1: *Supongamos que $G_1, G_2, \ldots, G_t$ son subespacios vectoriales de $\mathbb{F}_2^n$ de dimensión k tales que $G_i \cap G_j = \{\mathbf{0}\}$ para $i, j = 1, 2, \ldots, t$ con $i \neq j$, y consideremos el conjunto*

$$B = \begin{cases} \bigcup_{i=1}^{t} G_i^*, & \text{si } \ t = 2^{k-1}, \\ \{\mathbf{0}\} \cup \bigcup_{i=1}^{t} G_i^*, & \text{si } \ t = 2^{k-1}+1, \end{cases}$$

con $G_i^* = G_i \setminus \{\mathbf{0}\}$. Entonces B es el soporte de una función bent de n variables.

Dillon [12] denotó por $\mathcal{PS}^-$ (respectivamente, $\mathcal{PS}^+$), la clase de funciones bent para la que $t = 2^{k-1}$ (respectivamente, $t = 2^{k-1}+1$).

Ahora, construimos funciones bent basadas en la clase $\mathcal{PS}$ utilizando los complementos ortogonales de los subespacios vectoriales en lugar de los propios subespacios. Pero primero recordemos que el **complemento ortogonal** de un subespacio vectorial G de $\mathbb{F}_2^n$, denotado por $G^\perp$, es el subespacio vectorial

$$G^\perp = \{\boldsymbol{y} \in \mathbb{F}_2^n \mid \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0, \text{ para todo } \boldsymbol{x} \in G\}.$$

El siguiente resultado, cuya demostración se puede encontrar en cualquier libro de álgebra lineal, será necesario para probar el corolario 1, el cual establece que los complementos ortogonales de los subespacios vectoriales satisfacen las condiciones necesarias para que los soportes que definen correspondan a funciones bent de la clase $\mathcal{PS}$.

Lema 1: *Si G y H son dos subespacios vectoriales de $\mathbb{F}_2^n$ tales que $G \cap H = \{\mathbf{0}\}$ y $\dim G = \dim H = k$, entonces*

$$G^\perp \cap H^\perp = \{\mathbf{0}\} \quad y \quad \dim G^\perp = \dim H^\perp = k.$$

Ahora, como consecuencia inmediata del lema anterior, tenemos el siguiente resultado.

Corolario 1: *Sean $G_1, G_2, \ldots, G_t$ subespacios vectoriales de $\mathbb{F}_2^n$ de dimensión k, tales que $G_i \cap G_j = \{\mathbf{0}\}$ para $i, j = 1, 2, \ldots, t$ con $i \neq j$. Entonces,*

$$B^{(\perp)} = \begin{cases} \bigcup_{i=1}^{t} \left(G_i^\perp\right)^*, & \text{si } \ t = 2^{k-1}, \\ \{\mathbf{0}\} \cup \bigcup_{i=1}^{t} \left(G_i^\perp\right)^*, & \text{si } \ t = 2^{k-1}+1, \end{cases}$$

Demostración: Para $i = 1, 2, \ldots, 2^{k-1}+1$ tenemos que

$$\dim G_i^\perp = n - \dim G_i = k.$$

Además, si $\boldsymbol{x} \in G_i^\perp \cap G_j^\perp$, entonces

$$\langle \boldsymbol{x}, \boldsymbol{u} \rangle = \langle \boldsymbol{x}, \boldsymbol{v} \rangle = 0, \quad \text{para todo } \boldsymbol{u} \in G_i, \boldsymbol{v} \in G_j,$$

es decir,

$$\langle \boldsymbol{x}, \boldsymbol{u} \oplus \boldsymbol{v} \rangle = 0, \quad \text{para todo } \boldsymbol{u} \in G_i, \boldsymbol{v} \in G_j,$$

y por el lema 1,

$$\langle \boldsymbol{x}, \boldsymbol{w} \rangle = 0, \quad \text{para todo } \boldsymbol{w} \in \mathbb{F}_2^n.$$

Por tanto, $\boldsymbol{x} = \mathbf{0}$ y, en consecuencia $G_i^\perp \cap G_j^\perp = \{\mathbf{0}\}$.

Ahora, por el teorema 1 tenemos que $B^{(\perp)}$ es el soporte de una función bent de n variables. ■

III. Resultados principales

Ya estamos en condiciones de construir, de forma explícita, los subespacios G_i del teorema 1. Para ello, primero consideramos algunas bases del espacio columna de las matrices obtenidas a partir de dos matrices de tamaño $n \times k$ con coeficientes en $\mathbb{F}_2$ y cuyo rango es k; y la matriz asociada a un polinomio primitivo de grado k en $\mathbb{F}_2[\mathsf{X}]$.

La demostración del siguiente lema es directa y por ello la omitimos.

Lema 2: *Supongamos que C es la matriz asociada al polinomio primitivo*

$$c_0 + c_1\mathsf{X} + c_2\mathsf{X}^2 + \cdots + c_{k-1}\mathsf{X}^{k-1} + \mathsf{X}^k \in \mathbb{F}_2[\mathsf{X}].$$

Supongamos también que la matriz $U = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_k \end{bmatrix}$ de tamaño $n \times k$ tiene rango k, y consideremos, para $j = 1, 2, \ldots, 2^k - 2$, el vector

$$\boldsymbol{u}_{j+k} = c_0\boldsymbol{u}_j \oplus c_1\boldsymbol{u}_{j+1} \oplus \cdots \oplus c_{k-1}\boldsymbol{u}_{j+k-1} \in \mathbb{F}_2^n.$$

Entonces, para $i = 1, 2, \ldots, 2^k - 1$,

$$UC^{i-1} = \begin{bmatrix} \boldsymbol{u}_i & \boldsymbol{u}_{i+1} & \cdots & \boldsymbol{u}_{i+k-1} \end{bmatrix}$$

y

$$\mathrm{col}\,(U) = \mathrm{col}\left(UC^{i-1}\right).$$

Notemos que como consecuencia de este resultado, tenemos que

$$\mathrm{rg}\left(UC^{i-1}\right) = k, \quad \text{para } i = 1, 2, \ldots, 2^k - 1$$

y, por tanto, las columnas de la matriz UC^{i-1} constituyen también una base de $\mathrm{col}\,(U)$.

El teorema siguiente nos permite construir una familia de subespacios G_i que satisfacen las condiciones del teorema 1.

Teorema 2: *Sean U y V matrices de tamaño $n \times k$ tales que $\begin{bmatrix} U & V \end{bmatrix}$ es invertible y supongamos que C es la matriz asociada a un polinomio primitivo de grado k en $\mathbb{F}_2[\mathsf{X}]$. Si $G_0 = \operatorname{col}(V)$, $G_{2^k} = \operatorname{col}(U)$ y*

$$G_i = \operatorname{col}\left(UC^{i-1} \oplus V\right), \quad \textit{para } i = 1, 2, \ldots, 2^k - 1,$$

entonces $\dim G_r = k$, y $G_r \cap G_s = \{\mathbf{0}\}$ para $r, s = 0, 1, 2, \ldots, 2^k$ con $r \neq s$.

Demostración: Claramente $\operatorname{rg}(U) = \operatorname{rg}(V) = k$. Por tanto, $\dim G_0 = \dim G_{2^k} = k$.

Si $\dim G_i < k$ para algún $i \in \{1, 2, \ldots, 2^k - 1\}$, entonces, existe $\boldsymbol{a} \in \mathbb{F}_2^k \setminus \{\mathbf{0}\}$ tal que

$$\left(UC^{i-1} \oplus V\right)\boldsymbol{a} = UC^{i-1}\boldsymbol{a} \oplus V\boldsymbol{a} = \mathbf{0}$$

pero esto es una contradicción ya que las columnas de la matriz $\begin{bmatrix} UC^{i-1} & V \end{bmatrix}$ forman una base de $\mathbb{F}_2^n$ de acuerdo con la elección de las matrices U y V y el lema 2. En consecuencia, $\dim G_i = k$ para $i = 1, 2, \ldots, 2^k - 1$.

Evidentemente, $G_0 \cap G_{2^k} = \{\mathbf{0}\}$.

Supongamos en primer lugar que $\boldsymbol{w} \in G_0 \cap G_i$ para algún $i \in \{1, 2, 3, \ldots, 2^{k-1} - 1\}$. Entonces, de acuerdo con el lema 2

$$\boldsymbol{w} = \left(UC^{i-1} \oplus V\right)\boldsymbol{a} \quad \text{y} \quad \boldsymbol{w} = V\boldsymbol{b}$$

para algunos $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_2^k$. Por tanto,

$$UC^{i-1}\boldsymbol{a} \oplus V(\boldsymbol{a} \oplus \boldsymbol{b}) = \mathbf{0} \tag{1}$$

y otra vez, como las columnas de la matriz $\begin{bmatrix} UC^{i-1} & V \end{bmatrix}$ forman una base de $\mathbb{F}_2^n$, de la expresión (1) podemos decir que

$$\boldsymbol{a} = \mathbf{0} \quad \text{y} \quad \boldsymbol{a} \oplus \boldsymbol{b} = \mathbf{0};$$

así que, $\boldsymbol{b} = \mathbf{0}$ con lo que $\boldsymbol{w} = \mathbf{0}$. Por tanto, $G_0 \cap G_i = \{\mathbf{0}\}$ para $i \in \{1, 2, 3, \ldots, 2^k - 1\}$.

Supongamos ahora que $\boldsymbol{w} \in G_i \cap G_j$ para algunos i, j tales que $1 \leq i < j \leq 2^k - 1$. Procediendo como en el caso anterior, tenemos que

$$\boldsymbol{w} = \left(UC^{i-1} \oplus V\right)\boldsymbol{a} = \left(UC^{j-1} \oplus V\right)\boldsymbol{b}, \tag{2}$$

por tanto, de acuerdo con el lema 2, tenemos que

$$\begin{aligned} \mathbf{0} &= \left(UC^{i-1} \oplus V\right)\boldsymbol{a} \oplus \left(UC^{j-1} \oplus V\right)\boldsymbol{b} \\ &= U\boldsymbol{d} \oplus V(\boldsymbol{a} \oplus \boldsymbol{b}) \end{aligned}$$

para algún $\boldsymbol{d} \in \mathbb{F}_2^k$. Sin embargo, como las columnas de la matriz $\begin{bmatrix} U & V \end{bmatrix}$ forman una base de $\mathbb{F}_2^n$, necesariamente

$$\boldsymbol{d} = \boldsymbol{a} \oplus \boldsymbol{b} = \mathbf{0}.$$

En particular, $\boldsymbol{a} = \boldsymbol{b}$ y, sustituyendo en la expresión (2), obtenemos que

$$UC^{i-1}\boldsymbol{a} = UC^{j-1}\boldsymbol{a}$$

y, por tanto, $\left(C^{j-i} \oplus I\right)\boldsymbol{a} = \mathbf{0}$ ya que $\operatorname{rg}(U) = k$ e $i < j$. Ahora, como C es la matriz asociada a un polinomio primitivo de grado k y $\mathbb{F}_{2^k}$ es isomorfo a $\mathbb{F}_2[C]$ (véase [20]), podemos afirmar que la matriz $C^{j-i} \oplus I$ es invertible y, en consecuencia, $\boldsymbol{a} = \mathbf{0}$. Por tanto, sustituyendo dicho valor en la expresión (2), tenemos que $\boldsymbol{w} = \mathbf{0}$ y así, $G_i \cap G_j = \{\mathbf{0}\}$.

Finalmente, supongamos que $\boldsymbol{w} \in G_{2^k} \cap G_i$ para algún $i \in \{1, 2, 3, \ldots, 2^k - 1\}$. Entonces, procediendo como en el caso anterior, tenemos que

$$\boldsymbol{w} = U\boldsymbol{b} \quad \text{y} \quad \boldsymbol{w} = \left(UC^{i-1} \oplus V\right)\boldsymbol{a}, \tag{3}$$

para algunos $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_2^k$. Por tanto, de acuerdo con el lema 2, tenemos que

$$\mathbf{0} = \left(UC^{i-1} \oplus V\right)\boldsymbol{a} \oplus U\boldsymbol{b} = U\boldsymbol{d} \oplus V\boldsymbol{a}$$

para algún $\boldsymbol{d} \in \mathbb{F}_2^k$. Otra vez, como las columnas de la matriz $\begin{bmatrix} U & V \end{bmatrix}$ forman una base de $\mathbb{F}_2^n$, necesariamente $\boldsymbol{d} = \boldsymbol{a} = \mathbf{0}$, y, sustituyendo en la expresión (3), obtenemos que $\boldsymbol{w} = \mathbf{0}$. Por tanto, $G_{2^k} \cap G_i = \{\mathbf{0}\}$ para $i = 1, 2, \ldots, 2^k - 1$. ■

En el teorema 2, podríamos haber considerado también los subespacios vectoriales

$$F_j = \operatorname{col}\left(U \oplus VC^{j-1}\right) \quad \text{para } j = 1, 2, \ldots, 2^k - 1,$$

ya que satisfacen todas las condiciones necesarias siempre que $i + j \neq 2^k + 1$. Sin embargo, esto no es posible ya que cada uno de estos subespacios coincide con alguno de los subespacios G_i definidos en dicho teorema, como ponemos de manifiesto en el siguiente resultado.

Teorema 3: *Sean U y V matrices de tamaño $n \times k$ tales que $\begin{bmatrix} U & V \end{bmatrix}$ es invertible, supongamos que C es la matriz asociada a un polinomio primitivo de grado k en $\mathbb{F}_2[\mathsf{X}]$ y consideremos, para $i, j \in \{1, 2, \ldots, 2^k - 1\}$, los subespacios vectoriales*

$$G_i = \operatorname{col}\left(UC^{i-1} \oplus V\right) \quad \text{y} \quad F_j = \operatorname{col}\left(U \oplus VC^{j-1}\right).$$

Entonces para todo $i \in \{1, 2, \ldots, 2^k - 1\}$ existe un único $j \in \{1, 2, \ldots, 2^k - 1\}$ tal que $G_i = F_j$.

Demostración: Claramente $F_1 = G_1$.

Supongamos que $\boldsymbol{w} \in G_i$, para algún $i \in \{2, 3, \ldots, 2^k - 1\}$. Por el lema 2, tenemos que

$$\boldsymbol{w} = \left(UC^{i-1} \oplus V\right)\boldsymbol{a} \tag{4}$$

para algún $\boldsymbol{a} \in \mathbb{F}_2^k$. Sea $\boldsymbol{b} = C^{i-1}\boldsymbol{a}$, entonces, por ser C la matriz asociada a un polinomio primitivo de grado k, tenemos que

$$\boldsymbol{a} = C^{2^k - 1}\boldsymbol{a} = C^{2^k - i}C^{i-1}\boldsymbol{a} = C^{2^k - i}\boldsymbol{b}.$$

Sustituyendo el valor de $\boldsymbol{a}$, obtenido anteriormente, en la expresión (4) tenemos que

$$\boldsymbol{w} = \left(UC^{i-1} \oplus V\right)C^{2^k - i}\boldsymbol{b} = \left(U \oplus VC^{2^k - i}\right)\boldsymbol{b}$$

con lo que $\boldsymbol{w} \in F_j$, con $j = 2^k + 1 - i$ y así $G_i \subseteq F_j$. Ahora, como $\dim G_i = \dim F_j$, necesariamente $G_i = F_j$.

Evidentemente $j \in \{2, 3, \ldots, 2^k - 1\}$. Además, si existe $l \in \{2, 3, \ldots, 2^k - 1\}$ tal que $G_i = F_l$, entonces, necesariamente $l = j$. ■

Finalmente, como consecuencia de los teoremas 1 y 2 tenemos el siguiente resultado que nos permite construir el

soporte de una función bent a partir de una base de $\mathbb{F}_2^n$ y la matriz asociada a un polinomio primitivo de grado k en $\mathbb{F}_2[\mathsf{X}]$.

Corolario 2: *Supongamos que*

$$\mathcal{A} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k\}$$

es una base de $\mathbb{F}_2^n$ y consideremos las matrices

$$U = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_k \end{bmatrix} \quad y \quad V = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_k \end{bmatrix}.$$

Supongamos también que C es una matriz asociada a un polinomio primitivo de grado k en $\mathbb{F}_2[\mathsf{X}]$, y definamos los subespacios vectoriales G_i, para $i = 0, 1, 2, \ldots, 2^k$, como en el teorema 2. Si $I \subseteq \{0, 1, 2, \ldots, 2^k\}$ con $|I| = 2^{k-1}$ (respectivamente, $|I| = 2^{k-1} + 1$), entonces

$$B = \bigcup_{i \in I} G_i^* \quad \left(\text{respectivamente, } B = \{\mathbf{0}\} \cup \bigcup_{i \in I} G_i^* \right)$$

es el soporte de una función bent de n variables.

Ahora, con la notación del corolario 2, si $|I| = 2^{k-1}$ y $J = \{0, 1, 2, \ldots, 2^k\} \setminus I$, y denotamos por $f(\boldsymbol{x})$ y $g(\boldsymbol{x})$ las funciones bent cuyos soportes son los conjuntos

$$B_I = \bigcup_{i \in I} G_i^* \quad \text{y} \quad B_J = \{\mathbf{0}\} \cup \bigcup_{j \in J} G_j^*,$$

respectivamente, entonces $g(\boldsymbol{x}) = 1 \oplus f(\boldsymbol{x})$.

IV. Procedimiento práctico

A continuación, describimos un procedimiento práctico para obtener los subespacios G_i del teorema 2 a partir de una base $\mathcal{A}$ de $\mathbb{F}_2^n$ y la matriz C asociada a un polinomio primitivo de grado k en $\mathbb{F}_2[\mathsf{X}]$. Supongamos pues, como en el corolario 2, que agrupamos los vectores de $\mathcal{A}$ en dos matrices U y V de tamaño $n \times k$. Así, de acuerdo con lo dicho en la sección II, los vectores no nulos de G_0, G_i para $i = 1, 2, 3, \ldots, 2^k - 1$ y G_{2^k} son, respectivamente, las columnas de las matrices

$$V_0 = VH, \quad V_i = (UC^{i-1} \oplus V)H \quad \text{y} \quad V_{2^k} = UH$$

donde H es la matriz de control de paridad del $[2^k - 1, k]$-código binario de Hamming. Por tanto, si consideramos, por ejemplo, el conjunto $I = \{0, 1, 2, 3, \ldots, 2^{k-1} - 1\}$, entonces, de acuerdo con los teoremas 1 y 2 y lo dicho anteriormente, tenemos que las columnas de la matriz B siguiente, constituyen el soporte de una función bent de n variables.

$$\begin{aligned} B &= \begin{bmatrix} V_0 & V_1 & V_2 & V_3 & \cdots & V_{2^{k-1}-1} \end{bmatrix} \\ &= \Big[\; VH \quad (U \oplus V)H \quad (UC \oplus V)H \\ &\qquad (UC^2 \oplus V)H \quad \cdots \quad (UC^{2^{k-1}-2} \oplus V)H \; \Big] \end{aligned}$$

El siguiente ejemplo nos ayudará a entender el proceso anterior.

Ejemplo 1: Supongamos que $n = 6$ (y, por tanto, $k = 3$) y consideremos, por ejemplo, la base $\mathcal{A} = \{\mathbf{1}, \mathbf{2}, \mathbf{4}, \mathbf{8}, \mathbf{16}, \mathbf{32}\}$ y las matrices

$$U = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{4} \end{bmatrix} \quad \text{y} \quad V = \begin{bmatrix} \mathbf{8} & \mathbf{16} & \mathbf{32} \end{bmatrix}.$$

Aquí

$$\mathbf{1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{2} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{3} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad \ldots$$

y así sucesivamente; es decir, denotamos por $\boldsymbol{i}$ el vector columna correspondiente a la expansión binaria de 6 bits del entero i, para $i = 0, 1, 2, \ldots, 2^6 - 1$.

Consideremos el polinomio primitivo $p(\mathsf{X}) = 1 + \mathsf{X} + \mathsf{X}^3 \in \mathbb{F}_2[\mathsf{X}]$ y su matriz asociada

$$C = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Finalmente, consideremos la matriz de control de paridad

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

del $[7, 3]$-código binario de Hamming.

De acuerdo con el proceso anteriormente descrito, tenemos que

$$\begin{aligned} V_0 = VH &= \begin{bmatrix} \mathbf{8} & \mathbf{16} & \mathbf{32} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{32} & \mathbf{16} & \mathbf{48} & \mathbf{8} & \mathbf{40} & \mathbf{24} & \mathbf{56} \end{bmatrix}, \\ V_1 = (U \oplus V)H &= \begin{bmatrix} \mathbf{9} & \mathbf{18} & \mathbf{36} \end{bmatrix} H \\ &= \begin{bmatrix} \mathbf{36} & \mathbf{18} & \mathbf{54} & \mathbf{9} & \mathbf{45} & \mathbf{27} & \mathbf{63} \end{bmatrix}, \\ V_2 = (UC \oplus V)H &= \begin{bmatrix} \mathbf{10} & \mathbf{20} & \mathbf{35} \end{bmatrix} H \\ &= \begin{bmatrix} \mathbf{35} & \mathbf{20} & \mathbf{55} & \mathbf{10} & \mathbf{41} & \mathbf{30} & \mathbf{61} \end{bmatrix}, \\ V_3 = (UC^2 \oplus V)H &= \begin{bmatrix} \mathbf{12} & \mathbf{19} & \mathbf{38} \end{bmatrix} H \\ &= \begin{bmatrix} \mathbf{38} & \mathbf{19} & \mathbf{53} & \mathbf{12} & \mathbf{42} & \mathbf{31} & \mathbf{57} \end{bmatrix}. \end{aligned}$$

Por tanto, las columnas de la matriz

$$B = \begin{bmatrix} V_0 & V_1 & V_2 & V_3 \end{bmatrix}$$

son los elementos del soporte de una función bent de 6 variables.

Si obtenemos las $2^3 + 1 = 9$ matrices V_i, podemos construir $\binom{2^3+1}{2^{3-1}} = 126$ funciones bent pertenecientes a la clase $\mathcal{PS}^-$ y 126 funciones bent pertenecientes a la clase $\mathcal{PS}^+$, siendo todas ellas distintas. □

Fijada una base $\mathcal{A}$ de $\mathbb{F}_2^n$ y un polinomio primitivo de grado k y coeficientes en $\mathbb{F}_2$, podemos considerar $\binom{2^k+1}{2^{k-1}}$ subconjuntos I de $\{0, 1, 2, \ldots, 2^k\}$ de 2^{k-1} elementos y la misma cantidad de subconjuntos de $2^{k-1} + 1$ elementos. Por tanto, de acuerdo con el teorema 2, podemos construir $2\binom{2^k+1}{2^{k-1}}$ soportes de funciones bent, que serán distintos dos a dos ya que si $i, j \in \{0, 1, 2, \ldots, 2^k\}$ con $i \neq j$, entonces $V_i \neq V_j$.

V. Problemas abiertos

Ya hemos comentado al final de la sección anterior, que fijada una base de $\mathbb{F}_2^n$ y un polinomio primitivo de grado k y coeficientes en $\mathbb{F}_2$, de acuerdo con el teorema 2, podemos construir $2\binom{2^k+1}{2^{k-1}}$ soportes de funciones bent, que son distintos dos a dos. Surge ahora, de modo natural, la pregunta siguiente: ¿los $2\binom{2^k+1}{2^{k-1}}$ soportes de funciones bent construidos de acuerdo con el corolario 1 son distintos de los anteriores? El ejemplo siguiente pone de manifiesto que la respuesta no es necesariamente afirmativa.

Ejemplo 2: Supongamos que $n = 4$ (y, por tanto, $k = 2$) y consideremos, por ejemplo, la base $\mathcal{A} = \{\mathbf{1}, \mathbf{2}, \mathbf{4}, \mathbf{8}\}$ y las matrices $U = \begin{bmatrix}\mathbf{1} & \mathbf{2}\end{bmatrix}$ y $V = \begin{bmatrix}\mathbf{4} & \mathbf{8}\end{bmatrix}$. Consideremos el polinomio primitivo $p(\mathsf{X}) = 1 + \mathsf{X} + \mathsf{X}^2 \in \mathbb{F}_2[\mathsf{X}]$, su matriz asociada C, y la matriz de control de paridad H del $[3, 2]$-código binario de Hamming.

De acuerdo con lo dicho al inicio de la sección IV, tenemos que

$$\begin{aligned}
V_0 &= VH = \begin{bmatrix}\mathbf{4} & \mathbf{8}\end{bmatrix}\begin{bmatrix}0 & 1 & 1\\ 1 & 0 & 1\end{bmatrix} = \begin{bmatrix}\mathbf{8} & \mathbf{4} & \mathbf{12}\end{bmatrix},\\
V_1 &= (U \oplus V)H = \begin{bmatrix}\mathbf{10} & \mathbf{5} & \mathbf{15}\end{bmatrix},\\
V_2 &= (UC \oplus V)H = \begin{bmatrix}\mathbf{11} & \mathbf{6} & \mathbf{13}\end{bmatrix},\\
V_3 &= (UC^2 \oplus V)H = \begin{bmatrix}\mathbf{9} & \mathbf{7} & \mathbf{14}\end{bmatrix},\\
V_4 &= UH = \begin{bmatrix}\mathbf{2} & \mathbf{1} & \mathbf{3}\end{bmatrix}.
\end{aligned}$$

Por tanto, al aplicar el corolario 2 a cada uno de los subconjuntos I de $\{0, 1, 2, 3, 4\}$ de 2 elementos (análogamente para los de 3 elementos), obtenemos que cada uno de los conjuntos

$$\begin{aligned}
B_1 &= \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{8}, \mathbf{12}\}, \quad B_2 = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{5}, \mathbf{10}, \mathbf{15}\},\\
B_3 &= \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{6}, \mathbf{11}, \mathbf{13}\}, \quad B_4 = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{7}, \mathbf{9}, \mathbf{14}\},\\
B_5 &= \{\mathbf{4}, \mathbf{5}, \mathbf{8}, \mathbf{10}, \mathbf{12}, \mathbf{15}\}, \quad B_6 = \{\mathbf{4}, \mathbf{6}, \mathbf{8}, \mathbf{11}, \mathbf{12}, \mathbf{13}\},\\
B_7 &= \{\mathbf{4}, \mathbf{7}, \mathbf{8}, \mathbf{9}, \mathbf{12}, \mathbf{14}\}, \quad B_8 = \{\mathbf{5}, \mathbf{6}, \mathbf{10}, \mathbf{11}, \mathbf{13}, \mathbf{15}\},\\
B_9 &= \{\mathbf{5}, \mathbf{7}, \mathbf{9}, \mathbf{10}, \mathbf{14}, \mathbf{15}\}, \quad B_{10} = \{\mathbf{6}, \mathbf{7}, \mathbf{9}, \mathbf{11}, \mathbf{13}, \mathbf{14}\},
\end{aligned}$$

es el soporte de una función bent de la clase $\mathcal{PS}^-$ de 4 variables.

Ahora, si calculamos los complementos ortogonales de los subespacios vectoriales cuyos vectores no nulos son las columnas de las matrices V_i, es decir, de los subespacios G_i del teorema 2, obtenemos los subespacios ortogonales

$$\begin{aligned}
G_0^{(\perp)} &= \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}\}, \quad G_1^{(\perp)} = \{\mathbf{0}, \mathbf{5}, \mathbf{10}, \mathbf{15}\},\\
G_2^{(\perp)} &= \{\mathbf{0}, \mathbf{7}, \mathbf{9}, \mathbf{14}\}, \quad G_3^{(\perp)} = \{\mathbf{0}, \mathbf{6}, \mathbf{11}, \mathbf{13}\},\\
G_4^{(\perp)} &= \{\mathbf{0}, \mathbf{4}, \mathbf{8}, \mathbf{12}\}.
\end{aligned}$$

Puesto que

$$G_0^{(\perp)} = G_4, \quad G_1^{(\perp)} = G_1, \quad G_2^{(\perp)} = G_3, \quad G_3^{(\perp)} = G_2 \quad \text{y} \quad G_4^{(\perp)} = G_0,$$

es evidente que cada uno de los soportes construidos de acuerdo con el corolario 1 coincide con alguno de los soportes construidos de acuerdo con el teorema 1.

Consideremos ahora la base $\mathcal{A}' = \{\mathbf{9}, \mathbf{11}, \mathbf{12}, \mathbf{4}\}$ y las matrices $U' = \begin{bmatrix}\mathbf{9} & \mathbf{11}\end{bmatrix}$ y $V' = \begin{bmatrix}\mathbf{12} & \mathbf{4}\end{bmatrix}$.

Procediendo como en el caso anterior, tenemos que

$$\begin{aligned}
V_0' &= \begin{bmatrix}\mathbf{4} & \mathbf{12} & \mathbf{8}\end{bmatrix}, \quad V_1' = \begin{bmatrix}\mathbf{15} & \mathbf{5} & \mathbf{10}\end{bmatrix},\\
V_2' &= \begin{bmatrix}\mathbf{6} & \mathbf{7} & \mathbf{1}\end{bmatrix}, \quad V_3' = \begin{bmatrix}\mathbf{13} & \mathbf{14} & \mathbf{3}\end{bmatrix},\\
V_4' &= \begin{bmatrix}\mathbf{11} & \mathbf{9} & \mathbf{2}\end{bmatrix}
\end{aligned}$$

y, por tanto, cada uno de los conjuntos

$$\begin{aligned}
B_1' &= \{\mathbf{1}, \mathbf{2}, \mathbf{6}, \mathbf{7}, \mathbf{9}, \mathbf{11}\}, \quad B_2' = \{\mathbf{1}, \mathbf{3}, \mathbf{6}, \mathbf{7}, \mathbf{13}, \mathbf{14}\},\\
B_3' &= \{\mathbf{1}, \mathbf{4}, \mathbf{6}, \mathbf{7}, \mathbf{8}, \mathbf{12}\}, \quad B_4' = \{\mathbf{1}, \mathbf{5}, \mathbf{6}, \mathbf{7}, \mathbf{10}, \mathbf{15}\},\\
B_5' &= \{\mathbf{2}, \mathbf{3}, \mathbf{9}, \mathbf{11}, \mathbf{13}, \mathbf{14}\}, \quad B_6' = \{\mathbf{2}, \mathbf{4}, \mathbf{8}, \mathbf{9}, \mathbf{11}, \mathbf{12}\},\\
B_7' &= \{\mathbf{2}, \mathbf{5}, \mathbf{9}, \mathbf{10}, \mathbf{11}, \mathbf{15}\}, \quad B_8' = \{\mathbf{3}, \mathbf{4}, \mathbf{8}, \mathbf{12}, \mathbf{13}, \mathbf{14}\},\\
B_9' &= \{\mathbf{3}, \mathbf{5}, \mathbf{10}, \mathbf{13}, \mathbf{14}, \mathbf{15}\}, \quad B_{10}' = \{\mathbf{4}, \mathbf{5}, \mathbf{8}, \mathbf{10}, \mathbf{12}, \mathbf{15}\},
\end{aligned}$$

es el soporte de una función bent de la clase $\mathcal{PS}^-$ de 4 variables.

Ahora, si calculamos, como antes, los complementos ortogonales de los subespacios vectoriales cuyos vectores no nulos son las columnas de las matrices V_i', es decir, de los subespacios G_i' del teorema 2, obtenemos los subespacios ortogonales

$$\begin{aligned}
(G_0')^{(\perp)} &= \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}\}, \quad (G_1')^{(\perp)} = \{\mathbf{0}, \mathbf{5}, \mathbf{10}, \mathbf{15}\},\\
(G_2')^{(\perp)} &= \{\mathbf{0}, \mathbf{6}, \mathbf{8}, \mathbf{14}\}, \quad (G_3')^{(\perp)} = \{\mathbf{0}, \mathbf{7}, \mathbf{11}, \mathbf{12}\},\\
(G_4')^{(\perp)} &= \{\mathbf{0}, \mathbf{4}, \mathbf{9}, \mathbf{13}\},
\end{aligned}$$

que proporcionan los siguientes soportes de funciones bent de la clase $\mathcal{PS}^-$ de 4 variables:

$$\begin{aligned}
(B_1')^{(\perp)} &= \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{9}, \mathbf{13}\},\\
(B_2')^{(\perp)} &= \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{5}, \mathbf{10}, \mathbf{15}\},\\
(B_3')^{(\perp)} &= \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{6}, \mathbf{8}, \mathbf{14}\},\\
(B_4')^{(\perp)} &= \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{7}, \mathbf{11}, \mathbf{12}\},\\
(B_5')^{(\perp)} &= \{\mathbf{4}, \mathbf{5}, \mathbf{9}, \mathbf{10}, \mathbf{13}, \mathbf{15}\},\\
(B_6')^{(\perp)} &= \{\mathbf{4}, \mathbf{6}, \mathbf{8}, \mathbf{9}, \mathbf{13}, \mathbf{14}\},\\
(B_7')^{(\perp)} &= \{\mathbf{4}, \mathbf{7}, \mathbf{9}, \mathbf{11}, \mathbf{12}, \mathbf{13}\},\\
(B_8')^{(\perp)} &= \{\mathbf{5}, \mathbf{7}, \mathbf{10}, \mathbf{11}, \mathbf{12}, \mathbf{15}\},\\
(B_9')^{(\perp)} &= \{\mathbf{5}, \mathbf{6}, \mathbf{8}, \mathbf{10}, \mathbf{14}, \mathbf{15}\},\\
(B_{10}')^{(\perp)} &= \{\mathbf{6}, \mathbf{7}, \mathbf{8}, \mathbf{11}, \mathbf{12}, \mathbf{14}\}.
\end{aligned}$$

Notemos que en este caso, a diferencia de lo que ocurría en el caso anterior, ninguno de los conjuntos $(B_j')^{(\perp)}$ coincide con ninguno de los conjuntos B_i'. □

Tal como se desprende del ejemplo anterior, y en todos los ejemplos que hemos comprobado, existen bases para las que cada uno de los soportes B coincide con alguno de los soportes $B^{(\perp)}$ y existen bases para las que ninguno de los soportes B coincide con ninguno de los soportes $B^{(\perp)}$. Este hecho sugiere los problemas siguientes: ¿Fijada una base cualquiera, son estas las dos únicas situaciones posibles? En caso afirmativo, ¿bajo qué condiciones se da cada una de dichas situaciones?

Además, si observamos de nuevo el ejemplo 2, vemos que $B_5 = B'_{10}$, es decir, solamente uno de los soportes obtenidos con la base $\mathcal{A}$ coincide con uno de los soportes obtenidos con la base $\mathcal{A}'$. No ocurre lo mismo si consideramos las bases $\mathcal{A}$ y $\mathcal{A}'' = \{\mathbf{1}, \mathbf{2}, \mathbf{5}, \mathbf{10}\}$, ya que en este caso, todos los soportes obtenidos con la base $\mathcal{A}$ coinciden con los soportes obtenidos con la base $\mathcal{A}''$. De nuevo, en todos los ejemplos que hemos comprobado se da alguna de estas dos situaciones, lo cual sugiere los problemas siguientes: ¿Dadas dos bases cualquieras, son estas las dos únicas situaciones posibles? En caso afirmativo, ¿bajo qué condiciones se da cada una de dichas situaciones?

Otro problema que aparece en esta construcción es si influye o no el polinomio primitivo considerado. Es decir, ¿para una misma base y polinomios primitivos diferentes, podemos obtener idénticas funciones bent? ¿y si consideramos bases y polinomios diferentes?

Estos y otros problemas que puedan surgir de esta construcción se abordarán en futuros trabajos.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente subvencionado por los proyectos MTM2008-06674-C02-01 del Ministerio de Ciencia e Innovación del Gobierno de España y ACOMP/2010/039 de la Generalitat Valenciana. La investigación de Verónica Requena también ha sido financiada con una ayuda del Vicerrectorado de Investigación, Desarrollo e Innovación de la Universitat d'Alacant destinada a la formación de doctores.

REFERENCIAS

[1] C. M. Adams and S. E. Tavares, "Generating and counting binary bent sequences," *IEEE Transactions on Information Theory*, vol. 35, no. 6, pp. 1170–1173, 1990.

[2] ——, "Generating bent sequences," *Discrete Applied Mathematics*, vol. 39, pp. 155–159, 1992.

[3] C. M. Adams, "Constructing symmetric ciphers using the CAST design procedure," *Designs, Codes and Cryptography*, vol. 12, pp. 283–316, 1997.

[4] T. P. Berger, A. Canteaut, P. Charpin, and Y. Laigle-Chapuy, "On almost perfect nonlinear functions over $\mathbb{F}_2^n$," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4160–4170, 2006.

[5] A. Braeken, V. Nikov, S. Nikova, and B. Preneel, "On Boolean functions with generalized cryptographic properties," in *Progress in Cryptology – INDOCRYPT 2004*, ser. Lecture Notes in Computer Science, A. Canteaut and K. Viswanathan, Eds. Berlin: Springer-Verlag, 2004, vol. 3348, pp. 120–135.

[6] A. Canteaut and P. Charpin, "Decomposing bent functions," *IEEE Transactions on Information Theory*, vol. 49, no. 8, pp. 2004–2019, 2003.

[7] C. Carlet, "On the secondary constructions of resilient and bent functions," *Progress in Computer Science and Applied Logic*, vol. 23, pp. 3–28, 2004.

[8] ——, "On bent and highly nonlinear balanced/resilient functions and their algebraic immunities," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC-16)*, ser. Lecture Notes in Computer Science, M. Fossorier, H. Imai, S. Lin, and A. Poli, Eds. Berlin: Springer-Verlag, 2006, vol. 3857, pp. 1–28.

[9] C. Carlet and P. Guillot, "An alternate characterization of the bentness of binary functions, with uniqueness," *Designs, Codes and Cryptography*, vol. 14, pp. 133–140, 1998.

[10] C. Carlet and Y. Tarannikov, "Covering sequences of Boolean functions and their cryptographic significance," *Designs, Codes and Cryptography*, vol. 25, pp. 263–279, 2002.

[11] C. Charnes, M. Rötteler, and T. Beth, "Homogeneous bent functions, invariants, and designs," *Designs, Codes and Cryptography*, vol. 26, pp. 139–154, 2002.

[12] J. F. Dillon, "Elementary Hadamard difference sets," Ph.D. dissertation, University of Maryland, 1974.

[13] H. Dobbertin, "Construction of bent functions and balanced Boolean functions with high nonlinearity," in *Fast Sofware Encription*, ser. Lecture Notes in Computer Science, B. Preneel, Ed. Berlin: Springer-Verlag, 1995, vol. 1008, pp. 61–74.

[14] H. Dobbertin, G. Leander, A. Canteaut, C. Carlet, P. Felke, and P. Gaborit, "Construction of bent functions via Niho power functions," *Journal of Combinatorial Theory (Series A)*, vol. 113, no. 5, pp. 779–798, 2004.

[15] J. Fuller, E. Dawson, and W. Millan, "Evolutionary generation of bent functions for cryptography," in *Proceedings of the 2003 Congress on Evolutionary Computation*, vol. 2. IEEE, 2003, pp. 1655–1661.

[16] K. C. Gupta and P. Sarkar, "Improved construction of nonlinear resilient S-boxes," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 339–348, 2005.

[17] P. V. Kumar, R. A. Scholtz, and L. R. Welch, "Generalized bent functions and their properties," *Journal of Combinatorial Theory (Series A)*, vol. 40, pp. 90–107, 1985.

[18] K. Kurosawa and R. Matsumoto, "Almost security of cryptographic Boolean functions," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2752–2761, 2004.

[19] A. Lempel and M. Cohn, "Maximal families of bent sequences," *IEEE Transactions on Information Theory*, vol. 28, no. 6, pp. 865–868, 1982.

[20] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*. New York, NY: Cambridge University Press, 1997.

[21] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Advances in Cryptology – EUROCRYPT '93*, ser. Lecture Notes in Computer Science, T. Helleseth, Ed. Berlin: Springer-Verlag, 1994, vol. 765, pp. 386–397.

[22] R. L. McFarland, "A family of difference sets in non-cyclic groups," *Journal of Combinatorial Theory (Series A)*, vol. 15, pp. 1–10, 1973.

[23] W. Meier and O. Staffelbach, "Nonlinearity criteria for cryptographic functions," in *Advances in Cryptology – EUROCRYPT'89*, ser. Lecture Notes in Computer Science, J. Quisquater and J. Vandewalle, Eds. Berlin: Springer-Verlag, 1990, vol. 434, pp. 549–562.

[24] K. Nyberg, "Constructions of bent functions and difference sets," in *Advances in Cryptology – EUROCRYPT '90*, ser. Lecture Notes in Computer Science, I. B. Damgård, Ed. Berlin: Springer-Verlag, 1991, vol. 473, pp. 151–160.

[25] ——, "Perfect nonlinear S-boxes," in *Advances in Cryptology – EUROCRYPT '91*, ser. Lecture Notes in Computer Science, D. W. Davies, Ed. Berlin: Springer-Verlag, 1991, vol. 547, pp. 378–386.

[26] J. D. Olsen, R. A. Scholtz, and L. R. Welch, "Bent-function sequences," *IEEE Transactions on Information Theory*, vol. 28, no. 6, pp. 858–864, 1982.

[27] S. Roman, *Introduction to Coding and Information Theory*. New York, NY: Springer, 1997.

[28] O. S. Rothaus, "On "bent" functions," *Journal of Combinatorial Theory (Series A)*, vol. 20, pp. 300–305, 1976.

[29] P. Sarkar and S. Maitra, "Construction of nonlinear Boolean functions with important cryptographic properties," in *Advances in Cryptology – EUROCRYPT 2000*, ser. Lecture Notes in Computer Science, B. Preneel, Ed. Berlin: Springer-Verlag, 2000, vol. 1807, pp. 485–506.

[30] J. Seberry and X.-M. Zhang, "Highly nonlinear 0-1 balanced Boolean functions satisfying strict avalanche criterion (extended abstract)," in *Advances in Cryptology – ASIACRYPT'92*, ser. Lecture Notes in Computer Science, J. Seberry and Y. Zheng, Eds. Berlin: Springer-Verlag, 1992, vol. 718, pp. 145–155.

[31] J. Seberry, X.-M. Zhang, and Y. Zheng, "Nonlinearity and propagation characteristics of balanced Boolean functions," *Information and Computation*, vol. 119, pp. 1–13, 1995.

Características de Linealidad en Generadores de Secuencia Cifrante

A. Fúster Sabater
Instituto de Física Aplicada
C.S.I.C.
Serrano 144, 28006 Madrid
Email: amparo@iec.csic.es

P. Caballero Gil
DEIOC, Facultad de Matemáticas
Universidad de La Laguna
38271 Tenerife, Islas Canarias
Email: pcaballe@ull.es

***Abstract*—En este trabajo se muestra que las secuencias obtenidas a partir de un generador de secuencia cifrante, el Generalized Self-Shrinking Generator (GSSG), son soluciones particulares de una ecuación en diferencias lineal homogénea con coeficientes constantes. Además de las secuencias producidas por el GSSG, la clase completa de soluciones de dicha ecuación incluye otras secuencias equilibradas muy adecuadas para criptografía, pues tienen el mismo periodo e incluso mayor complejidad lineal que las producidas por el GSSG. Los parámetros criptográficos de todas las secuencias anteriormente mencionadas pueden analizarse en términos de soluciones de ecuaciones en diferencias lineales.**

I. Introducción

Los generadores de secuencias pseudoaleatorias tiene numerosas aplicaciones en comunicaciones seguras, por ejemplo en redes inalámbricas, debido a ventajas prácticas tales como facilidad de implementación, alta velocidad y fiabilidad. A partir de una clave corta, un generador de secuencia binaria produce una serie larga de bits pseudoaleatorios denominada *secuencia cifrante*. La mayor parte de los generadores de secuencia cifrante están basados en Linear Feedback Shift Registers (LFSRs) [5] cuyas secuencias de salida, las *PN*-secuencias, se combinan de forma no lineal para producir una secuencia pseudoaleatoria de aplicación criptográfica. Generadores combinacionales, filtros no-lineales, generadores controlados por reloj o decimados de forma irregular son algunos de los ejemplos más conocidos que pueden encontrarse en la literatura [9], [13].

Coppersmith, Krawczyk y Mansour [1] propusieron el *Shrinking Generator*, luego Meier y Staffelbach [12] diseñaron el *Self-Shrinking Generator*, finalmente Hu y Xiao [8] definieron el *Generalized Self-Shrinking Generator* (GSSG) que genera una familia de secuencias pseudoaleatorias (que denominaremos *secuencias generalizadas*) muy adecuadas para uso criptográfico por tener largos periodos, buena correlación, excelente distribución de rachas, equilibrio entre ceros y unos, simplicidad de implementación etc. El GSSG puede considerarse como una especialización del generador shrinking a la vez que una generalización del generador self-shrinking. De hecho, la secuencia de salida del generador self-shrinking es un elemento de la familia de secuencias producidas por el GSSG. La idea fundamental de los generadores de este tipo es la decimación irregular de una *PN*-secuencia según determinen los bits de otra. El resultado de esa decimación es la secuencia cifrante. Entre los criptoanálisis más destacados de generadores basados en decimación pueden reseñarse [3], [4], [7], [14], [15] y [16].

El trabajo aquí realizado se resume en dos ideas básicas:

1) Se muestra que las secuencias generadas por un GSSG son soluciones particulares de un tipo de ecuación lineal en diferencias.
2) Se muestra que algunas secuencias soluciones de la ecuación, aunque no estén incluidas en la familia producida por el GSSG, sí presentan buenas propiedades criptográficas. En general, las soluciones de las ecuaciones lineales en diferencias proporcionan un método sencillo para la generación de secuencias pseudoaleatorias.

II. El Generalized Self-Shrinking Generator (GSSG)

El GSSG es un generador de secuencia binaria que puede describirse tal y como sigue:

1) Hace uso de dos secuencias: una *PN*-secuencia $\{a_n\}$ y una versión desplazada de la misma denotada por $\{v_n\}$.
2) Relaciona ambas secuencias mediante una simple regla de decimación para generar la secuencia de salida.

El resultado de los pasos anteriores es una familia de *secuencias generalizadas* que puede definirse de una manera más formal [8] como:

Definición 1: Sea $\{a_n\}$ una *PN*-secuencia de periodo 2^L-1 generada a partir de un LFSR con polinomio característico primitivo y grado L. Sea G un vector binario de dimensión L definido como:

$$G = (g_0, g_1, g_2, \ldots, g_{L-1}) \in GF(2)^L. \quad (1)$$

El *n*-ésimo elemento de la secuencia $\{v_n\}$ se define como:

$$v_n = g_0 a_n \oplus g_1 a_{n-1} \oplus g_2 a_{n-2} \oplus \ldots \oplus g_{L-1} a_{n-L+1}, \quad (2)$$

donde los subíndices de la secuencia $\{a_n\}$ se calculan mod $2^L - 1$ y el símbolo $\oplus$ representa la operación lógica OR-exclusiva. Para $n \geq 0$ la regla de decimación es muy sencilla:

1) Si $a_n = 1$, entonces v_n es bit de la secuencia de salida.

2) Si $a_n = 0$, entonces v_n se descarta y no hay bit de salida.

De esta forma, se genera la secuencia de salida $b_0\, b_1\, b_2 \ldots$ notada $\{b_n\}$ ó $\{b(G)\}$. Dicha secuencia es una secuencia generalizada o secuencia asociada a un valor de G.

Nótese que la secuencia $\{v_n\}$ no es más que una versión desplazada de la secuencia $\{a_n\}$. Cuando G recorre el conjunto $GF(2)^L - (0, \ldots, 0)$, entonces $\{v_n\}$ corresponde a los $2^L - 1$ posibles desplazamientos de $\{a_n\}$. Además, el conjunto de secuencias denotado por $B(a) = \{\{b(G)\}, G \in GF(2)^L\}$ es la familia de secuencias generalizadas basadas en la *PN*-secuencia $\{a_n\}$.

Ejemplo 1. Para la *PN*-secuencia $\{a_n\} = \{11110101100\ 1000\}$ con polinomio de realimentación x^4+x+1, se obtienen 16 secuencias generalizadas basadas en la *PN*-secuencia $\{a_n\}$ (véase [8]):

1. $G = (0000), \{b(G)\} = 00000000 \sim$
2. $G = (1000), \{b(G)\} = 11111111 \sim$
3. $G = (0100), \{b(G)\} = 01110010 \sim$
4. $G = (1100), \{b(G)\} = 10001101 \sim$
5. $G = (0010), \{b(G)\} = 00111100 \sim$
6. $G = (1010), \{b(G)\} = 11000011 \sim$
7. $G = (0110), \{b(G)\} = 01001110 \sim$
8. $G = (1110), \{b(G)\} = 10110001 \sim$
9. $G = (0001), \{b(G)\} = 00011011 \sim$
10. $G = (1001), \{b(G)\} = 11100100 \sim$
11. $G = (0101), \{b(G)\} = 01101001 \sim$
12. $G = (1101), \{b(G)\} = 10010110 \sim$
13. $G = (0011), \{b(G)\} = 00100111 \sim$
14. $G = (1011), \{b(G)\} = 11011000 \sim$
15. $G = (0111), \{b(G)\} = 01010101 \sim$
16. $G = (1111), \{b(G)\} = 10101010 \sim$

Es importante señalar que las secuencias generadas no son 16 secuencias diferentes. De hecho, las secuencias 5 y 6 son versiones desplazadas de una misma secuencia y lo mismo sucede con las secuencias 11 y 12 y con la 15 y 16. Al mismo tiempo, las secuencias 3, 7, 10 y 13 corresponden a una única secuencia al igual que las secuencias 4, 8, 9 y 14. Diferentes parámetros critográficos de estas secuencias tales como periodo o complejidad lineal se analizan en las siguientes secciones en términos de soluciones de ecuaciones en diferencias lineales.

III. Ecuaciones en Diferencias Lineales con Coeficientes Constantes

En este trabajo se considera el siguiente tipo de ecuación en diferencias con coeficientes binarios:

$$(E^r \oplus \sum_{j=1}^{r} c_j\ E^{r-j})\ z_n = 0, \qquad n \geq 0 \tag{3}$$

donde $z_n \in GF(2)$ es el *n*-ésimo término de una secuencia binaria $\{z_n\}$ que verifica la ecuación anterior. El símbolo E es el operador desplazamiento que actúa sobre los elementos z_n de la secuencia solución, es decir $E^j z_n = z_{n+j}$. Los coeficientes c_j son valores binarios $c_j \in GF(2)$ y r es un número entero. El polinomio característico de grado r de la ecuación (3) es:

$$P(x) = x^r + \sum_{j=1}^{r} c_j\ x^{r-j}, \tag{4}$$

y especifica la relación de recurrencia lineal en $\{z_n\}$. Es decir, el *n*-ésimo término, z_n, puede escribirse como una combinación lineal de los r términos precedentes:

$$z_n \oplus \sum_{j=1}^{r} c_j\ z_{n-j} = 0, \qquad n \geq r. \tag{5}$$

Si $P(x)$ es un polinomio primitivo [10] y α es una de sus raíces, entonces

$$\alpha,\ \alpha^2,\ \alpha^{2^2}, \ldots,\ \alpha^{2^{(r-1)}} \in GF(2)^r \tag{6}$$

son las r raíces diferentes de dicho polinomio (véase [11]). En este caso, se demuestra en [10] que la solución de (3) es una secuencia de la forma:

$$z_n = \sum_{j=0}^{r-1} A^{2^j}\ \alpha^{2^j n}, \qquad n \geq 0 \tag{7}$$

donde A es un elemento arbitrario de $GF(2)^r$. Es decir, $\{z_n\}$ es una *PN*-secuencia de polinomio característico $P(x)$ y periodo $2^r - 1$ cuyo inicio está determinado por el valor de A.

Las ecuaciones en diferencias consideradas anteriormente pueden generalizarse a un tipo más complejo de ecuaciones cuyas raíces tengan una multiplicidad mayor que 1. De hecho, se consideran ecuaciones de la forma:

$$(E^r \oplus \sum_{j=1}^{r} c_j\ E^{r-j})^p\ z_n = 0, \qquad n \geq 0 \tag{8}$$

siendo p un entero $p > 1$. El polinomio característico de este tipo de ecuación es:

$$P_M(x) = P(x)^p = (x^r + \sum_{j=1}^{r} c_j\ x^{r-j})^p. \tag{9}$$

Ahora las raíces de $P_M(x)$ son las mismas que las del polinomio $P(x)$, es decir, $(\alpha, \alpha^2, \alpha^{2^2}, \ldots,\ \alpha^{2^{(r-1)}})$, pero con multiplicidad p. Entonces las soluciones de (8) son [2]:

$$z_n = \sum_{i=0}^{p-1} \binom{n}{i} \sum_{j=0}^{r-1} A_i^{2^j}\ \alpha^{2^j n}), \qquad n \geq 0 \tag{10}$$

donde $A_i \in GF(2)^r$ y los números combinatorios $\binom{n}{i}$ se calculan mod 2.

En resumen, el *n*-ésimo término $\{z_n\}$ de una secuencia solución es la suma bit a bit del *n*-ésimo término de p secuencias $\{\sum_{j=0}^{r-1} A_i^{2^j}\ \alpha^{2^j n}\}$ $(0 \leq i < p)$ ponderadas por coeficientes que son números combinatorios.

De hecho, cuando n va tomando sucesivos valores cada número combinatorio $\binom{n}{i}$ $(n \geq i \geq 0)$ define una *secuencia*

TABLE I
NÚMEROS COMBINATORIOS, SECUENCIAS PRIMARIAS Y PERIODOS T_i

Núm. combinator.	Secuencias primarias	T_i
$\binom{n}{0}$	$S_0 = \{1,1,1,1,1,1,1,1 \sim\}$	$T_0 = 1$
$\binom{n}{1}$	$S_1 = \{0,1,0,1,0,1,0,1 \sim\}$	$T_1 = 2$
$\binom{n}{2}$	$S_2 = \{0,0,1,1,0,0,1,1 \sim\}$	$T_2 = 4$
$\binom{n}{3}$	$S_3 = \{0,0,0,1,0,0,0,1 \sim\}$	$T_3 = 4$
$\binom{n}{4}$	$S_4 = \{0,0,0,0,1,1,1,1 \sim\}$	$T_4 = 8$
$\binom{n}{5}$	$S_5 = \{0,0,0,0,0,1,0,1 \sim\}$	$T_5 = 8$
$\binom{n}{6}$	$S_6 = \{0,0,0,0,0,0,1,1 \sim\}$	$T_6 = 8$
$\binom{n}{7}$	$S_7 = \{0,0,0,0,0,0,0,1 \sim\}$	$T_7 = 8$

primaria con periodo constante T_i. Se asume que $\binom{n}{i} = 0$ para $n < i$. En la Tabla I, aparecen representados los primeros números combinatorios con sus correspondientes secuencias primarias y periodos. Se aprecia que la generación de dichas secuencias sigue una regla sencilla. En efecto, para $i = 0$ la secuencia generada es la idénticamente 1, mientras que las 2^m secuencias primarias asociadas con $\binom{n}{i}$ para $(2^m \leq i < 2^{m+1})$ (siendo m un entero no negativo) tienen periodo $T_i = 2^{m+1}$ y sus dígitos son:

1) Los primeros 2^m bits son ceros.
2) Los siguientes 2^m bits son los primeros 2^m bits de la secuencia $\binom{n}{i-2^m}$.

Lo vemos con un simple ejemplo. De acuerdo con la Tabla I y para $m = 2$, tenemos 2^2 secuencias primarias S_i con $(2^2 \leq i < 2^3)$. La secuencia S_4 tiene 2^2 ceros y los 2^2 primeros dígitos de S_0. De la misma forma, la secuencia S_5 tiene 2^2 ceros y los 2^2 primeros dígitos de S_1. Igualmente, la secuencia S_6 tiene 2^2 ceros y los 2^2 primeros dígitos de S_2 mientras que S_7 tiene 2^2 ceros y los 2^2 primeros dígitos de S_3. Por tanto, la generación y manipulación de tales secuencias en muy fácil.

IV. PRINCIPAL RESULTADO

Ahora se presenta el resultado fundamental que relaciona secuencias generalizadas con ecuaciones en diferencias.

Teorema 1: La familia de secuencias generalizadas $B(a)$ basada en la *PN*-secuencia $\{a_n\}$ son soluciones particulares de la ecuación en diferencias lineal homogénea:

$$(E \oplus 1)^p \; z_n = 0, \qquad p = 2^{L-1} \; , \tag{11}$$

con polinomio característico $(x+1)^p$.

Idea de la demostración: Como los periodos de las secuencias $B(a)$ son $T \in \{1, 2, 2^{L-1}\}$ [8], entonces el periodo T de cualquier secuencia es divisor de 2^{L-1}, por tanto $x^T + 1 = (x+1)^T$. Por otro lado, si $f(x)$ es el polinomio característico de la menor relación de recurrencia verificada por la secuencia generalizada, entonces la condición $f(x)|x^T + 1$ implica que $f(x)$ es de la forma:

$$f(x) = (x+1)^{LC} \tag{12}$$

donde LC es la complejidad lineal de dicha secuencia. Por tanto las secuencias generalizadas verifican (11) y son soluciones particulares de dicha ecuación lineal en diferencias. □

Ahora podemos pues analizar al detalle las características de las secuencias que verifican la ecuación en diferencias previa. Según (10), las soluciones de la ecuación dada en (11) son ahora de la forma:

$$z_n = \binom{n}{0} A_0 \oplus \binom{n}{1} A_1 \oplus \ldots \oplus \binom{n}{p-1} A_{p-1}, \qquad n \geq 0 \tag{13}$$

donde $A_i \in GF(2)$ son coeficientes binarios, $\alpha = 1$ es la única raíz del polinomio $(x+1)$ de grado $r = 1$ con multiplicidad p y $\binom{n}{i}$ $(0 \leq i < p)$ son números combinatorios mod 2. Nótese que la secuencia $\{z_n\}$ es simplemente una suma lógica bit a bit de secuencias primarias ponderadas por los correspondientes coeficientes A_i. Así pues, diferentes elecciones de A_i darán lugar a diferentes secuencias con distintas características. A partir de la ecuación (13) se pueden analizar aspectos particulares de las secuencias solución. Todos ellos están relacionados con la elección de la *p*-tupla $(A_0, A_1, A_2, \ldots, A_{p-1})$ de coeficientes binarios.

A. Periodos de las Secuencias Solución

Según la sección 3, los periodos de las secuencias primarias son potencias de 2. Más aún, según (13) la secuencia $\{z_n\}$ es la suma lógica bit a bit de secuencias de diferentes periodos. Luego, el periodo de $\{z_n\}$ es el máximo periodo de las secuencias que aparecen en (13). De hecho, el period de $\{z_n\}$ es el T_i correspondiente a la secuencia primaria con el mayor índice i tal que $A_i \neq 0$.

B. Complejidad Lineal de las Secuencias Solución

Según [10], la complejidad lineal de una secuencia coincide con el número de raíces de su polinomio característico que aparecen en su relación de recurrencia lineal. Según esto y analizando los coeficientes A_i en (13), se puede calcular el valor de la complejidad lineal. De hecho, tenemos una única raíz con multiplicidad p. Luego, si i es el mayor índice $(0 \leq i < p)$ para el que $A_i \neq 0$, entonces la complejidad lineal es LC de la secuencia $\{z_n\}$ será:

$$LC = i + 1 \tag{14}$$

puesto que será la multiplicidad de la raíz 1.

C. Número de Diferentes Secuencias Solución

Para contar el número de secuencias diferentes $\{z_n\}$ que son soluciones de (11), volvemos igualmente a considerar los coeficientes A_i en (13). Si i $(0 \leq i < p)$ es el mayor subíndice para el que $A_i \neq 0$, entonces hay 2^i posibles elecciones de la *i*-tupla $(A_0, A_1, A_2, , A_{i-1})$ para la secuencia $\{z_n\}$ en (13). Por otro lado, como el periodo de tal secuencia es T_i, el número de secuencias diferentes N_i será:

$$N_i = 2^i / T_i \qquad (0 \leq i < p). \tag{15}$$

El número total N_{total} de secuencias solución de la ecuación lineal en diferencias (11) será:

$$N_{total} = \sum_{i=0}^{p-1} N_i. \tag{16}$$

En resumen, la elección de los coeficientes A_i permite generar secuencias binarias con periodo y complejidad lineal controlables.

V. Un Ejemplo Ilustrativo

Volvemos al GSSG presentado en la sección 2. Para la *PN*-secuencia $\{a_n\} = \{111101011001000\}$ con polinomio primitivo de grado 4, la familia de secuencias generalizadas $B(a)$ son soluciones de:

$$(E \oplus 1)^p \, b_n = 0, \qquad p = 2^3 \, , \tag{17}$$

cuya forma general es:

$$b_n = \binom{n}{0} A_0 \oplus \binom{n}{1} A_1 \oplus \ldots \oplus \binom{n}{7} A_7, \qquad n \geq 0 \tag{18}$$

Podemos considerar diferentes elecciones de la 8-tupla $(A_0, A_1, \ldots, A_7)$:

1) Para $A_i = 0 \ \forall \, i$, la secuencia solución $\{b_n\} = \{0\}$ es la secuencia idénticamente nula incluída ya en la familia de secuencias generalizadas.
$$G = (0000), \{b(G)\} = 00000000 \sim .$$
2) Para $A_0 \neq 0, \ A_i = 0 \ \forall \, i > 0$, la secuencia solución $\{b_n\} = \{1111 \sim\}$ es la secuencia idénticamente 1 que está incluída en la familia de secuencias generalizadas:
$$G = (1000), \{b(G)\} = 11111111 \sim .$$
Una secuencia con $T_0 = 1$ y $LC_0 = 1$.
3) Para $A_1 \neq 0, \ A_i = 0 \ \forall \, i > 1$, existe una única secuencia solución $\{b_n\}$ con period $T_1 = 2$ y $LC_1 = 2$. El par $(A_0 = 0, \, A_1 = 1)$ genera $\{b_n\} = \{01 \sim\}$ que corresponde a la secuencia generalizada:
$$G = (0111), \{b(G)\} = 01010101 \sim .$$
El par $(A_0 = 1, \, A_1 = 1)$ genera $\{b_n\} = \{10 \sim\}$ que corresponde a la secuencia generalizada:
$$G = (1111), \{b(G)\} = 10101010 \sim .$$
4) Para $A_2 \neq 0, \ A_i = 0 \ \forall \, i > 2$, existe una única y equilibrada secuencia solución $\{b_n\}$ con periodo $T_2 = 4$ y $LC_2 = 3$. Por ejemplo, la terna $(A_0 = 0, A_1 = 0, A_2 = 1)$ genera $\{b_n\} = \{0011 \sim\}$. Otras ternas con $A_2 = 1$ dan lugar a versiones desplazadas de la misma secuencia pero ninguna de ellas es una secuencia generalizada.
5) Para $A_3 \neq 0, \ A_i = 0 \ \forall \, i > 3$, existen 2 secuencias no equilibradas con periodo $T_3 = 4$ y $LC_3 = 4$. Al ser secuencias no equilibradas, ninguna de ellas pertenece a la familia de secuencias generalizadas.
6) Para $A_4 \neq 0, \ A_i = 0 \ \forall \, i > 4$, hay 2 secuencias equilibradas diferentes con periodo $T_4 = 8$ y $LC_4 = 5$. Por ejemplo, la 5-tupla $(A_0 = 0, A_1 = 0, A_2 = 1, A_3 = 0, A_4 = 1)$ genera $\{b_n\} = \{00111100 \sim\}$ que corresponde a la secuencia generalizada:
$$G = (0010), \{b(G)\} = 00111100 \sim .$$
Mientras que la 5-tupla $(A_0 = 0, A_1 = 1, A_2 = 1, A_3 = 0, A_4 = 1)$ genera $\{b_n\} = \{01101001 \sim\}$ que corresponde a la secuencia generalizada:
$$G = (0101), \{b(G)\} = 01101001 \sim .$$
7) Para $A_5 \neq 0, \ A_i = 0 \ \forall \, i > 5$, hay 4 secuencias diferentes no todas equilibradas con periodo $T_5 = 8$ y $LC_5 = 6$. Por ejemplo, la 6-tupla $(A_0 = 0, A_1 = 1, A_2 = 1, A_3 = 1, A_4 = 0, A_5 = 1)$ genera $\{b_n\} = \{01110010 \sim\}$ que corresponde a la secuencia generalizada:
$$G = (0100), \{b(G)\} = 01110010 \sim .$$
8) Para $A_6 \neq 0, \ A_i = 0 \ \forall \, i > 6$, hay 8 secuencias diferentes no todas equilibradas con periodo $T_6 = 8$ y $LC_6 = 7$. Ninguna de ellas corresponde a secuencias generalizadas.
Sin embargo existen 4 secuencias solución equilibradas $\{b_n\} = \{01010110 \sim\}$, $\{b_n\} = \{10101001 \sim\}$, $\{b_n\} = \{01011100 \sim\}$ and $\{b_n\} = \{10100011 \sim\}$ con el mismo periodo, autocorrelación y mayor complejidad lineal que las secuencias generalizadas descritas en los pasos 6 y 7.
9) Finalmente para $A_7 \neq 0, \ A_i = 0 \ \forall \, i > 7$, hay 16 secuencias diferentes no equilibradas con periodo $T_7 = 8$ y $LC_7 = 8$. Ninguna de ellas corresponde a secuencias generalizadas.

En resumen, podemos concluir:

1) La selección de coeficientes A_i permite controlar el periodo, la complejidad y el equilibrio entre ceros y unos de las secuencias solución.
2) Todas las secuencias generalizadas, las 7 secuencias distintas del ejemplo anterior, aparecen como soluciones de la ecuación lineal en diferencias (17).

VI. Conclusiones

En este trabajo se muestra que la familia de secuencias generalizadas, y por tanto las secuencia producida por el generador auto-shrinking, son soluciones particulares de ecuaciones en diferencias lineales. Al mismo tiempo, existen otras muchas secuencias solución no incluídas en la familia anterior que presentan las mismas o incluso mejores características criptográficas. Más aún, la elección de la *p*-tupla $(A_0, A_1, A_2, \ldots, A_{p-1})$ de coeficientes binarios permite:

1) Obtener todas las soluciones de la ecuación en diferencias referenciada anteriormente (13), entre las que se encuentran secuencias con aplicación en cifrado en flujo.
2) Generar secuencias cuyo periodo, complejidad lineal y equilibrio entre ceros y unos sea controlable.

Hay que notar que aunque las secuencias generalizadas se obtienen por decimación irregular a partir de LFSRs, en la práctica son simples soluciones de ecuaciones lineales. Este hecho establece una sutil relación entre decimación irregular

y linealidad que puede ser convenientemente explotada en el criptanálisis de tales generadores de secuencia. Una prolongación natural de este trabajo es la generalización de estos resultados a las llamadas secuencias *interleaved* [6], puesto que presentan una estructura muy similar a las secuencias obtenidas mediante decimación irregular.

AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente por el CDTI y las empresas INDRA, Unión Fenosa, Tecnobit, Visual Tool, Brainstorm, SAC y Technosafe en el marco del Proyecto Cenit-HESPERIA; también ha sido financiado por el Ministerio de Ciencia e Innovación y el fondo europeo FEDER en el marco del Proyecto TIN2008-02236/TSI.

REFERENCES

[1] D. Coppersmith, H. Krawczyk and Y. Mansour, The Shrinking Generator. Proc. of CRYPTO'93. Lecture Notes in Computer Science, Springer Verlag, Vol. 773, pp. 22-39, 1994.

[2] L. E. Dickson, Linear Groups with an Exposition of the Galois Field Theory. New York: Dover, 1958, pp. 3-71. An updated reprint can be found at http://www-math.cudenver.edu/ wcherowi/courses/finflds.html

[3] A. Fúster-Sabater and P. Caballero-Gil, Strategic Attack on the Shrinking Generator, Theoretical Computer Science, Vol. 409, No. 3, pp. 530-536, December 2008.

[4] A. Fúster-Sabater and P. Caballero-Gil, Cryptanalytic Attack on Cryptographic Sequence Generators: The Class of Clock-Controlled Shrinking Generators. Proc. of ICCSA 2008, Part II. Lecture Notes on Computer Science, Springer-Verlag, Vol. 5073, pp. 668-679, 2008.

[5] S.W. Golomb, Shift Register-Sequences, Aegean Park Press, Laguna Hill, 1982.

[6] G. Gong, Theory and Applications of q-ary Interleaved Sequences, IEEE Trans. Information Theory, Vol. 41, No. 2, pp. 400-411, 1995.

[7] M. Gomulkiewicz, M. Kutylowski and P. Wlaz, Fault Jumping Attacks against Shrinking Generator, Dagstuhl Seminar, 2006, Proceedings 06111, Complexity of Boolean Functions, available at http://drops.dagstuhl.de/opus/volltexte/2006/611

[8] Y. Hu and G. Xiao, Generalized Self-Shrinking Generator, IEEE Trans. Inform. Theory, Vol. 50, pp. 714-719, April 2004.

[9] S.M. Jennings, Multiplexed Sequences: Some Properties, in Proc. EUROCRYPT'83. Lecture Notes in Computer Science, Springer Verlag, Vol. 149, 1983.

[10] E.L. Key, An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators, IEEE Trans. Informat. Theory, Vol. 22, No. 6, pp. 732-736, 1976.

[11] R. Lidl and H. Niederreiter, Introduction to Finite Fields and Their Applications, Cambridge, England: Cambridge University Press, 1986.

[12] W. Meier and O. Staffelbach, The Self-Shrinking Generator, in Proc. EUROCRYPT94. Lecture Notes in Computer Science, Springer Verlag, Vol. 950, pp. 205-214, 1995.

[13] A.J. Menezes *et al.*, Handbook of Applied Cryptography, New York:CRC Press, 1997.

[14] M. J. Mihaljevic, A Faster Cryptanalysis of the Self-Shrinking Generator, in Proc. ACISP96. Lecture Notes in Computer Science, Springer Verlag, Vol. 1172, pp. 182-189, 1996.

[15] E. Zenner, M. Krause and S. Lucks, Improved cryptanalysis of the self-shrinking generator, in Proc. ACISP01. Lecture Notes in Computer Science, Springer Verlag, Vol. 2119, pp. 21-35, 2001.

[16] B. Zhang and D. Feng, New Guess-and-Determine Attack on the Self-Shrinking Generator, in Proc. ASIACRYPT06. Lecture Notes in Computer Science, Springer Verlag, Vol. 4284, pp. 54-68, 2006.

Estudio de las Propiedades de Propagación de la Divergencia de los Autómatas Celulares Elementales

Ángel Martín del Rey
Departamento de Matemática Aplicada
E.P.S. de Ávila
Universidad de Salamanca
Email: delrey@usal.es

Araceli Queiruga Dios
Departamento de Matemática Aplicada
E.T.S.I.I. de Béjar
Universidad de Salamanca
Email: queirugadios@usal.es

Gerardo Rodríguez Sánchez
Departamento de Matemática Aplicada
E.P.S. de Zamora
Universidad de Salamanca
Email: gerardo@usal.es

***Abstract*—Como es bien conocido, los autómatas celulares elementales no cumplen de manera satisfactoria muchas de las propiedades criptográficas requeridas para su uso directo en el desarrollo de criptosistemas de clave secreta. Ello es debido fundamentalmente a la sencillez de las funciones booleanas en las que se basan sus reglas de transición local. En el presente trabajo se muestra cómo construir a partir de los autómatas celulares elementales autómatas más complejos mediante la introducción del concepto de divergencia del autómata celular elemental. Se estudian además las principales propiedades de propagación de la divergencia.**

I. Introducción

Han sido numerosos los trabajos publicados sobre los usos criptográficos de los autómatas celulares tanto en la criptografía de clave secreta como en la criptografía de clave pública (véase, por ejemplo, [1], [2], [7], [8]). Los autómatas celulares (AC para abreviar) están íntimamente relacionados con las funciones booleanas ya que son dichas funciones las que rigen su evolución cuando el conjunto de estados es $\mathbb{F}_2$. En consecuencia, el estudio de las aplicaciones criptográficas de los AC está muy ligado al estudio de las propiedades criptográficas de las funciones booleanas que definen sus funciones de transición local (equilibrio, características de propagación, no linealidad, resistencia, etc.).

Un tipo particularmente interesante de AC son los autómatas celulares elementales. Se trata de AC cuyo conjunto de estados es $\mathbb{F}_2$ y cuya función de transición local viene definida por una función booleana de 3 variables. Debido a la simplicidad de estas funciones, las aplicaciones criptográficas de este tipo de AC son limitadas.

El objetivo principal de este trabajo es la búsqueda de un método simple que nos permita construir AC más complejos a partir de los autómatas celulares elementales y que posean mejores propiedades de carácter criptográfico. En este sentido se introduce la noción de divergencia de un autómata celular elemental y se estudian algunas de sus propiedades criptográficas: el equilibrio y las características de propagación.

El resto del trabajo se organiza como sigue: en la sección 2 se introduce la teoría básica sobre las funciones booleanas y los autómatas celulares elementales; la divergencia de los autómatas celulares elementales se define en la sección 3, y algunas de sus propiedades criptográficas son analizadas en la sección 4. Finalmente, en la sección 5 se presentan las conclusiones.

II. Preliminares Matemáticos

A. Funciones booleanas

Sea $\mathbb{F}_2^n$ el espacio vectorial n-dimensional sobre el cuerpo de Galois $\mathbb{F}_2 = \{0, 1\}$, y sea $\{e_1, \ldots, e_n\}$ su base canónica. Una función booleana (en n variables) es una aplicación de la forma $f : \mathbb{F}_2^n \to \mathbb{F}_2$, siendo $\mathcal{BF}_n$ el conjunto de las mismas; obsérvese que su cardinal es $|\mathcal{BF}_n| = 2^{2^n}$.

Se denomina peso de Hamming del vector $u = (u_1, \ldots, u_n) \in \mathbb{F}_2^n$ y se denota por $wt(u)$ al número de sus coordenadas no nulas. Por otra parte, el peso de Hamming de la función booleana $f \in \mathcal{BF}_n$ se define como

$$wt(f) = |\{u \in \mathbb{F}_2^n \text{ tal que } f(u) \neq 0\}|. \tag{1}$$

Es más, la distancia de Hamming entre dos funciones booleanas $f, g \in \mathcal{BF}_n$ se define como $d(f, g) = wt(f \oplus g)$, donde $(f \oplus g)(u) = f(u) \oplus g(u)$.

La representación más común de las funciones boolenas es mediante su Forma Normal Algebraica (FNA), que no es más que su representación polinómica sobre $\mathbb{F}_2$, es decir:

$$f(u_1, \ldots, u_n) = a_0 \oplus \bigoplus_{\substack{1 \leq k \leq n \\ 1 \leq i_1, i_2, \ldots, i_k \leq n}} a_{i_1 i_2 \ldots i_k} u_{i_1} u_{i_2} \ldots u_{i_k}, \tag{2}$$

donde $a_0, a_{i_1 \ldots i_k} \in \mathbb{F}_2$. Se llama grado de la FNA al grado algebraico del polinomio.

Una función vectorial booleana es una aplicación de la forma:

$$\begin{aligned} F : \mathbb{F}_2^n &\to \mathbb{F}_2^m \\ u &\mapsto F(u) = (F_1(u), \ldots, F_m(u)) \end{aligned} \tag{3}$$

donde $F_i : \mathbb{F}_2^n \to \mathbb{F}_2$ son funciones booleanas en n variables.

La derivada parcial de una función booleana $f \in \mathcal{BF}_n$ con respecto a la i-ésima variable u_i es otra función booleana $D_i f \in \mathcal{BF}_n$ definida como sigue:

$$\begin{aligned} D_i f : \mathbb{F}_2^n &\to \mathbb{F}_2 \\ u &\mapsto D_i f(u) = f(u) \oplus f(u \oplus e_i) \end{aligned} \tag{4}$$

El concepto de derivada parcial se puede extender al concepto de derivada direccional de la siguiente manera: la derivada direccional de $f \in \mathcal{BF}_n$ con respecto al vector $b \in \mathbb{F}_2^n$ es

$$\begin{array}{rcl} D_b f \colon \mathbb{F}_2^n & \to & \mathbb{F}_2 \\ u & \mapsto & D_b f(u) = f(u) \oplus f(u \oplus b) \end{array} \quad (5)$$

Obsérvese que la derivada parcial respecto a la variable u_i no es más que la derivada direccional con respecto al vector $e_i \in \mathbb{F}_2^n$, esto es $D_i f = D_{e_i} f$. Además se verifica el siguiente importante resultado (véase [5]):

Teorema 1: Sea $f \in \mathcal{BF}_n$ una función booleana y consideremos $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ con $k \leq n$, entonces:

$$\left(D_{e_{i_1}} \circ \ldots \circ D_{e_{i_k}}\right) f = \bigoplus_{\substack{1 \leq l \leq k \\ j_1 < \ldots < j_l \\ j_1, \ldots, j_l \in \{i_1, \ldots, i_k\}}} D_{e_{j_1} \oplus \ldots \oplus e_{j_l}} f. \quad (6)$$

B. Autómatas Celulares

Los AC (véase [11]) son máquinas de estados finitos constituidas por m unidades de memoria denominadas células que se disponen una a continuación de otra a modo de cadena. En cada instante de tiempo cada célula adopta un estado perteneciente al conjunto finito de estados $\mathbb{F}_2$, de modo que el estado de la célula i-ésima en el instante de tiempo t se denota por $x_i^t \in \mathbb{F}_2$. Los estados de las diferentes células van cambiando en cada instante (discreto) de tiempo de manera sincronizada y de acuerdo a una determinada función de transición local f. Dicha función es una función booleana de k variables, con $k \leq m$, las cuales no son más que los estados de las células vecinas en el instante inmediatamente anterior de tiempo, es decir:

$$\begin{array}{rcl} f \colon \mathbb{F}_2^k & \to & \mathbb{F}_2 \\ \left(x_{i+\alpha_1}^t, \ldots, x_{i+\alpha_k}^t\right) & \mapsto & x_i^{t+1} = f\left(x_{i+\alpha_1}^t, \ldots, x_{i+\alpha_k}^t\right) \end{array} \quad (7)$$

para todo $1 \leq i \leq m$. Consecuentemente, existen 2^{2^k} posibles autómatas celulares.

Obsérvese que los índices $V = \{\alpha_1, \ldots, \alpha_k\} \subset \mathbb{Z}$ definen la vecindad de cada célula del autómata celular (en este trabajo supondremos que las vecindades son homogéneas). En este sentido, si $V = \{-q, \ldots, 0, \ldots, q\}$ se dice que el autómata celular posee vecindades simétricas de radio q.

Como el número de células es finito, se deben establecer condiciones de contorno para asegurar que la evolución del AC se realice de manera correcta. Normalmente se consideran condiciones de contorno periódicas:

$$x_i^t = x_j^t \text{ si } i \equiv j \pmod{m} \text{ para todo } t. \quad (8)$$

El vector m-dimensional $X^t = (x_1^t, \ldots, x_m^t) \in \mathbb{F}_2^m$ se denomina configuración del AC en el instante de tiempo t. La dinámica completa de un AC viene definida por la llamada función de transición global:

$$\begin{array}{rcl} \Phi \colon \mathbb{F}_2^m & \to & \mathbb{F}_2^m \\ X^t & \mapsto & \Phi\left(X^t\right) = X^{t+1} \end{array} \quad (9)$$

Obsérvese que Φ viene definida por la siguiente función vectorial booleana:

$$\Phi(u) = (\Phi_1(u), \ldots, \Phi_m(u)), \quad (10)$$

donde

$$\Phi_i(u) = f(u_{i+\alpha_1}, \ldots, u_{i+\alpha_k}),\ 1 \leq i \leq m, \quad (11)$$

es una función booleana en k variables.

Un tipo particular y muy interesante de AC lo forman los Autómatas Celulares Elementales (ACE para abreviar). Estos autómatas poseen vecindades simétricas de radio $q = 1$, con lo que existen $2^{2^3} = 256$ de los mismos. Cada uno de ellos tiene asociado un número w que se puede calcular como sigue (véase [10]):

$$0 \leq w = \sum_{i=0}^{7} \alpha_i \cdot 2^i \leq 255, \quad (12)$$

donde la tabla de verdad de la función booleana f es la siguiente:

s_{i-1}^t	s_i^t	s_{i+1}^t	$\mapsto$	s_i^{t+1}
0	0	0	$\mapsto$	α_0
0	0	1	$\mapsto$	α_1
0	1	0	$\mapsto$	α_2
0	1	1	$\mapsto$	α_3
1	0	0	$\mapsto$	α_4
1	0	1	$\mapsto$	α_5
1	1	0	$\mapsto$	α_6
1	1	1	$\mapsto$	α_7

III. Divergencia de un ACE

En esta sección introduciremos el concepto de divergencia de un ACE y algunos resultados de interés para este trabajo:

Definición 1: La divergencia de un autómata celular elemental cuyo espacio celular está formado por m células y cuya función de transición global viene dada por (9)-(11) es un autómata celular con $p > m$ células, con vecindad

- Si m es impar con $m = 2r + 1$, entonces:

$$V = \{-\frac{m-1}{2}, \ldots, 0, \ldots, \frac{m-1}{2}\}. \quad (13)$$

- Si m es par con $m = 2r$, entonces:

$$V = \{-\frac{m}{2} + 1, \ldots, 0, \ldots, \frac{m}{2}\}. \quad (14)$$

y cuya función de transición local es

$$\begin{array}{rcl} \operatorname{div}(\Phi) : \mathbb{F}_2^m & \to & \mathbb{F}_2 \\ u & \mapsto & \operatorname{div}(\Phi)(u) = \bigoplus_{i=1}^{m} D_i \Phi_i(u) \end{array} \quad (15)$$

Proposición 1: La divergencia de un ACE arbitrario cuya función de transición local viene dada por

$$\begin{array}{rcl} f\left(x_{i-1}^t, x_i^t, x_{i+1}^t\right) & = & a_0 \oplus a_1 x_{i-1}^t \oplus a_2 x_i^t \oplus a_3 x_{i+1}^t \\ & & \oplus a_{12} x_{i-1}^t x_i^t \oplus a_{13} x_{i-1}^t x_{i+1}^t \\ & & \oplus a_{23} x_i^t x_{i+1}^t \oplus a_{123} x_{i-1}^t x_i^t x_{i+1}^t, \end{array} \quad (16)$$

es

$$\begin{aligned}\operatorname{div}(\Phi)\left(x_{i-r}^t\ldots,x_{i+r}^t\right) &= a_2 \oplus (a_{12}\oplus a_{23}) \bigoplus_{j=-r}^{r} x_{i+j}^t \\ &\oplus a_{123}x_{i-r}^t x_{i+r-1}^t \\ &\oplus a_{123}x_{i-r+1}^t x_{i+r}^t \\ &\oplus a_{123}\bigoplus_{j=-r}^{r-2} x_{i+j}^t x_{i+j+2}^t. \quad (17)\end{aligned}$$

si $m = 2r+1$ es impar, y

$$\begin{aligned}\operatorname{div}(\Phi)\left(x_{i-r+1}^t\ldots,x_{i+r}^t\right) &= (a_{12}\oplus a_{23}) \bigoplus_{j=-r+1}^{r} x_{i+j}^t \\ &\oplus a_{123}x_{i-r+1}^t x_{i+r-1}^t \\ &\oplus a_{123}x_{i-r+2}^t x_{i+r}^t \\ &\oplus a_{123}\bigoplus_{j=-r+1}^{r-2} x_{i+j}^t x_{i+j+2}^t. \quad (18)\end{aligned}$$

si $m = 2r$ es par.

Obsérvese que la divergencia de un ACE viene definida por medio de tres parámetros: $a_2, a_{12}\oplus a_{23}$ y a_{123} cuando m es impar, y por sólo dos parámetros, $a_{12}\oplus a_{23}$ y a_{123}, cuando m es par. Consecuentemente, existen $2^3 = 8$ posibles divergencias para cada m impar y $2^2 = 4$ divergencias distintas cuando m es par. Esas divergencias se muestran en la Tabla I y en la Tabla II, respectivamente.

Este resultado permite clasificar los autómatas celulares elementales en diferentes clases de acuerdo a la divergencia. Esta clasificación se muestra en las Tablas III y IV.

IV. Propiedades de propagación de la divergencia

En esta sección estudiaremos algunas de las principales propiedades criptográficas que se deben satisfacer para poder usar la divergencia en el desarrollo de criptosistemas de clave secreta. Concretamente centraremos nuestra atención tanto en el estudio del equilibrio (*balancedness* en inglés) como en las propiedades de propagación.

A. Equilibrio

Como es bien sabido, las funciones booleanas de uso en criptografía deben ser equilibradas, es decir, las salidas producidas deben estar uniformemente distribuidas sobre $\mathbb{F}_2$ (véase, por ejemplo [3]). En este sentido si f es una función booleana en n variables entonces es equilibrada o balanceada cuando $wt(f) = 2^{n-1}$. Esta propiedad permite evitar la dependencia estadística que puede existir entre las entradas y las salidas de la función booleana y que es aprovechada por cierto tipo de ataques criptoanalíticos.

Los ACE balanceados son los siguientes (véase [4]): 15, 23, 27, 29, 30, 39, 43, 45, 46, 51, 53, 54, 57, 58, 60, 71, 75, 77, 78, 83, 85, 86, 89, 90, 92, 99, 101, 102, 105, 106, 108, 113, 114, 116, 120, 135, 139, 141, 142, 147, 149, 150, 153, 154, 156, 163, 165, 166, 169, 170, 172, 177, 178, 180, 184, 195, 197, 198, 201, 202, 204, 209, 210, 212, 216, 225, 226, 228, 232, y 240.

TABLE I
Expresiones de las divergencias cuando m es impar

Parámetros	Divergencia
Clase I $a_2 = 0$ $a_{12}\oplus a_{23} = 0$ $a_{123} = 0$	0
Clase II $a_2 = 0$ $a_{12}\oplus a_{23} = 0$ $a_{123} = 1$	$x_{i-r}^t x_{i+r-1}^t \oplus x_{i-r+1}^t x_{i+r}^t \oplus \bigoplus_{j=-r}^{r-2} x_{i+j}^t x_{i+j+2}^t$
Clase III $a_2 = 0$ $a_{12}\oplus a_{23} = 1$ $a_{123} = 0$	$\bigoplus_{j=-r}^{r} x_{i+j}^t$
Clase IV $a_2 = 0$ $a_{12}\oplus a_{23} = 1$ $a_{123} = 1$	$\bigoplus_{j=-r}^{r} x_{i+j}^t \oplus x_{i-r}^t x_{i+r-1}^t \oplus x_{i-r+1}^t x_{i+r}^t \oplus \bigoplus_{j=-r}^{r-2} x_{i+j}^t x_{i+j+2}^t$
Clase V $a_2 = 1$ $a_{12}\oplus a_{23} = 0$ $a_{123} = 0$	1
Clase VI $a_2 = 1$ $a_{12}\oplus a_{23} = 0$ $a_{123} = 1$	$1 \oplus x_{i-r}^t x_{i+r-1}^t \oplus x_{i-r+1}^t x_{i+r}^t \oplus \bigoplus_{j=-r}^{r-2} x_{i+j}^t x_{i+j+2}^t$
Clase VII $a_2 = 1$ $a_{12}\oplus a_{23} = 1$ $a_{123} = 0$	$1 \oplus \bigoplus_{j=-r}^{r} x_{i+j}^t$
Clase VIII $a_2 = 1$ $a_{12}\oplus a_{23} = 1$ $a_{123} = 1$	$1 \oplus \bigoplus_{j=-r}^{r} x_{i+j}^t \oplus x_{i-r}^t x_{i+r-1}^t \oplus x_{i-r+1}^t x_{i+r}^t \oplus \bigoplus_{j=-r}^{r-2} x_{i+j}^t x_{i+j+2}^t$

TABLE II
Expresiones de la divergencia cuando m es par

Parámetros	Divergencia
Clase I $a_{12}\oplus a_{23} = 0$ $a_{123} = 0$	0
Clase II $a_{12}\oplus a_{23} = 0$ $a_{123} = 1$	$x_{i-r+1}^t x_{i+r-1}^t \oplus x_{i-r+2}^t x_{i+r}^t \oplus \bigoplus_{j=-r+1}^{r-2} x_{i+j}^t x_{i+j+2}^t$
Clase III $a_{12}\oplus a_{23} = 1$ $a_{123} = 0$	$\bigoplus_{j=-r+1}^{r} x_{i+j}^t$
Clase IV $a_{12}\oplus a_{23} = 1$ $a_{123} = 1$	$\bigoplus_{j=-r+1}^{r} x_{i+j}^t \oplus x_{i-r+1}^t x_{i+r-1}^t \oplus x_{i-r+2}^t x_{i+r}^t \oplus \bigoplus_{j=-r+1}^{r-2} x_{i+j}^t x_{i+j+2}^t$

En el caso que nos ocupa, el de las divergencias de los ACE, se obtienen los siguientes resultados:

Si m es impar:

- Las clases I y V no son equilibradas ya que las divergencias son funciones booleanas constantes. Además las clases IV y VIII tampoco son equilibradas.
- Las clases III y VII son equilibradas ya que las divergencias son funciones booleanas afines no constantes. Además las clases II y VI son también balanceadas.

TABLE III
CLASIFICACIÓN DE LOS ACE DE ACUERDO A SU DIVERGENCIA CUANDO m ES IMPAR

Clase	ACE
Clase I	0,5,10,15,18,23,24,29,66,71,72,77,80,85,90,95,160, 165,170,175,178,183,184,189,226,231,232,237,240, 245,250,255
Clase II	32,37,42,47,50,55,56,61,98,103,104,109,112,117, 122,127,128,133,138,143,146,151,152,157,194,199, 200,205,208,213,218,223
Clase III	34,39,40,45,48,53,58,63,96,101,106,111,114,119, 120,125,130,135,136,141,144,149,154,159,192,197, 202,207,210,215,216,221
Clase IV	2,7,8,13,16,21,26,31,64,69,74,79,82,87,88,93,162, 167,168,173,176,181,186,191,224,229,234,239,242, 247,248,253
Clase V	33,36,43,46,51,54,57,60,99,102,105,108,113,116, 123,126,129,132,139,142,147,150,153,156,195,198, 201,204,209,212,219,222
Clase VI	1,4,11,14,19,22,25,28,67,70,73,76,81,84,91,94,161, 164,171,174,179,182,185,188,227,230,233,236,241, 244,251,254
Clase VII	3,6,9,12,17,20,27,30,65,68,75,78,83,86,89,92,163, 166,169,172,177,180,187,190,225,228,235,238,243, 246,249,252
Clase VIII	35,38,41,44,49,52,59,62,97,100,107,110,115,118, 121,124,131,134,137,140,145,148,155,158,193,196, 203,206,211,214,217,220

TABLE IV
CLASIFICACIÓN DE LOS ACE DE ACUERDO A SU DIVERGENCIA CUANDO m ES PAR

Clase	ACE
Clase I	0,5,10,15,18,23,24,29,33,36,43,46,51,54,57,60, 66,71,72,77,80,85,90,95,99,102,105,108,113,116, 123,126,129,132,139,142,147,150,153,156,160, 165,170,175,178,183,184,189,195,198,201,204, 209,212,219,222,226,231,232,237,240,245,250,255
Clase II	1,4,11,14,19,22,25,28,32,37,42,47,50,55,56,61, 67,70,73,76,81,84,91,94,98,103,104,109,112,117, 122,127,128,133,138,143,146,151,152,157,161, 164,171,174,179,182,185,188,194,199,200,205, 208,213,218,223,227,230,233,236,241,244,251,254
Clase III	3,6,9,12,17,20,27,30,34,39,40,45,48,53,58,63, 65,68,75,78,83,86,89,92,96,101,106,111,114,119, 120,125,130,135,136,141,144,149,154,159,163, 166,169,172,177,180,187,190,192,197,202,207, 210,215,216,221,225,228,235,238,243,246,249,252
Clase IV	2,7,8,13,16,21,26,31,35,38,41,44,49,52,59,62, 64,69,74,79,82,87,88,93,97,100,107,110,115,118, 121,124,131,134,137,140,145,148,155,158,162, 167,168,173,176,181,186,191,193,196,203,206, 211,214,217,220,224,229,234,239,242,247,248,253

Si m es par:

- La clase I no es equilibrada ya que la divergencia es la función booleana nula.
- La clase III es equilibrada puesto que la divergencia es una función booleana afín (no constante).
- Si además m no es múltiplo de 4 entonces la clase II es equilibrada (en otro caso -m par y múltiplo de 4-, la clase II es no equilibrada).
- Si $m = 4, 12, 20 \ldots$ la clase IV es equilibrada (si $m \neq 4, 12, \ldots$ no es equilibrada).

B. Propiedades de propagación

En esta sección estudiaremos el Criterio Estricto de Avalancha (*Strict Avalanche Criterion* en inglés) y su generalización: el Criterio de Propagación (*Propagation Criterion* en inglés).

1) Criterio Estricto de Avalancha: Se dice que una función booleana satisface el Criterio Estricto de Avalancha (utilizaremos su acrónimo en inglés para abreviarlo: SAC) cuando la salida de la función cambia con una probabilidad de 0.5 siempre que se complemente un sólo bit de la entrada. Este criterio se puede caracterizar en términos de la derivada booleana como sigue (véase [9]): la función booleana f satisface el SAC si $D_j f$ es una función equilibrada para toda j. En el caso de la divergencia, se verifica lo siguiente:

Si m es par:

- Las derivadas parciales de las divergencias correspondientes a las clases I y III no son equilibradas puesto que se trata de las funciones booleanas constantes respectivamente:

$$D_{e_j}\mathrm{div}\,(\Phi) = 0, \quad (19)$$
$$D_{e_j}\mathrm{div}\,(\Phi) = 1. \quad (20)$$

  Por otra parte, las derivadas parciales de las divergencias de las clases II y IV son:

$$D_{e_j}\mathrm{div}\,(\Phi) = u_{j-2} \oplus u_{j+2}, \quad (21)$$
$$D_{e_j}\mathrm{div}\,(\Phi) = 1 \oplus u_{j-2} \oplus u_{j+2}, \quad (22)$$

  respectivamente (los subíndices se toman módulo m). Consecuentemente son equilibradas.

Si m es impar:

- Las derivadas parciales de las divergencias de las clases I y V no son equilibradas ya que se trata de la función constante nula:

$$D_{e_j}\mathrm{div}\,(\Phi) = 0. \quad (23)$$

- Las derivadas parciales de las divergencias de las clases III y VII tampoco son balanceadas puesto que se trata también de funciones booleanas constantes:

$$D_{e_j}\mathrm{div}\,(\Phi) = 1, \quad (24)$$

- Las divergencias cuyas derivadas parciales son equilibradas son las correspondientes a las clases II, VI y IV, VIII ya que un simple cálculo nos muestra que:

$$D_{e_j}\mathrm{div}\,(\Phi) = u_{j-2} \oplus u_{j+2}, \quad (25)$$
$$D_{e_j}\mathrm{div}\,(\Phi) = 1 \oplus u_{j-2} \oplus u_{j+2}, \quad (26)$$

  respectivamente (recuérdese que los subíndices se toman módulo m).

Por lo tanto, cuando m es par, sólo las clases II y IV son SAC, mientras que cuando m es impar, las clases que son SAC son la II, la IV, la VI y la VIII.

2) Criterio de Propagación: Para asegurar unas buenas propiedades de difusión, las funciones booleanas con aplicación en la criptografía deben satisfacer el Criterio de Propagación (utilizaremos su acrónimo en inglés, PC -*Propagation Criterion*- para hacer referencia a él a partir de ahora). Este criterio fue introducido por B. Preneel *et al.* (véase [6]) y está basado en las propiedades de las derivadas booleanas que muestran el comportamiento de las funciones estudiadas cuando algunas variables de la entrada son complementadas. Así, una función booleana de n variables, f, satisface el PC con respecto al subconjunto $B \subset \mathbb{F}_2^n$ si para todo $b \in B$ la derivada direccional $D_b f$ es equilibrada. Es más, la función booleana f satisface el PC de grado k (f es PC(k)) si dicha función satisface el PC respecto al siguiente conjunto:

$$\begin{aligned} W(k) &= \{b \in \mathbb{F}_2^n - \{0\} \text{ tal que } wt(b) \leq k\} \\ &= \left\{e_{i_1} \oplus \ldots \oplus e_{i_j},\, 1 \leq i_1 < \ldots < i_j \leq n, 1 \leq j \leq k\right\}. \end{aligned} \tag{27}$$

Obsérvese que el SAC no es más que el PC(1).

En el caso que nos ocupa en este trabajo, se verifica lo siguiente:

Las clases I y V cuando m es impar y la clase I cuando es par no satisfacen el PC ya que sus derivadas direccionales son la función booleana constante nula.

Las clases III y VII cuando m es impar y la clase III cuando m es par tampoco satisfacen el PC ya que las divergencias correspondientes son funciones booleanas afines (no constantes) y sus derivadas direccionales son funciones booleanas constantes (y consiguientemente, no equilibradas).

Las clases II y VI cuando m es impar, y la clase II cuando m es par satisfacen el PC, lo cual puede ser demostrado por recurrencia sobre k:

- Si $k = 1$: Dado que las clases II y VI son SAC, entonces son PC(1).
- Si $k = 2$: Teniendo en cuenta el Teorema 1, se verifica:

$$\begin{aligned} D_{e_{i_1} \oplus e_{i_2}} f &= \left(D_{e_{i_1}} \circ D_{e_{i_2}}\right) f \oplus D_{e_{i_1}} f \oplus D_{e_{i_2}} f \\ &= D_{e_{i_1}} \left(u_{i_2-2} \oplus u_{i_2+2}\right) \\ &\quad \oplus u_{i_1-2} \oplus u_{i_1+2} \oplus u_{i_2-2} \oplus u_{i_2+2} \\ &= \begin{cases} 1 \oplus u_{i_1-4} \oplus u_{i_1-2} \oplus u_{i_1} \oplus u_{i_1+2}, & \text{si } i_1 = i_2 - 2 \\ 1 \oplus u_{i_1-2} \oplus u_{i_1} \oplus u_{i_1+2} \oplus u_{i_1+4}, & \text{si } i_1 = i_2 + 2 \\ u_{i_1-2} \oplus u_{i_1+2} \oplus u_{i_2-2} \oplus u_{i_2+2}, & \text{en otro caso} \end{cases} \end{aligned} \tag{28}$$

  para todo $1 \leq i_1 < i_2 \leq n$. Entonces $D_{e_{i_1} \oplus e_{i_2}} f$ es una función booleana afín no constante y consecuentemente es equilibrada y es PC(2).
- Supongamos que la función es PC($k-1$), entonces:

$$\begin{aligned} D_{e_{i_1} \oplus \ldots \oplus e_{i_k}} f &= \left(D_{e_{i_1}} \circ \ldots \circ D_{e_{i_k}}\right) f \\ &\quad \oplus \bigoplus_{\substack{1 \leq l \leq k-1 \\ j_1 < \ldots < j_l \\ j_1, \ldots, j_l \in \{i_1, \ldots, i_k\}}} D_{e_{j_1} \oplus \ldots \oplus e_{j_l}} f. \end{aligned} \tag{29}$$

  Aplicando la recurrencia sabemos que cada sumando de la forma $D_{e_{j_1} \oplus \ldots \oplus e_{j_l}} f$ es una función booleana afín no constante. Además $\left(D_{e_{i_1}} \circ \ldots \circ D_{e_{i_k}}\right) f = 0$ para $k > 2$; en consecuencia $D_{e_{i_1} \oplus \ldots \oplus e_{i_k}} f$ es el sumatorio XOR de funciones booleanas afines no constantes y , por lo tanto, es equilibrado y satisface pues el PC(k).

Un argumento similar prueba que las clases IV y VIII con m impar, y la clase IV con m par satisfacen el PC.

V. Conclusiones

Como es bien sabido, la evolución de los autómatas celulares elementales viene regida, en último extremo, por una función booleana de 3 variables. La simplicidad de estas funciones es uno de los principales motivos que hacen que no se utilicen directamente los ACE en el desarrollo de criptosistemas de clave secreta. No obstante, es posible construir autómatas celulares más complejos a partir de los ACE. En este sentido se ha introducido en este trabajo el concepto de divergencia de un ACE (con un espacio celular formado por m células), cuya función de transición local viene definida por una función booleana de m variables. De manera más concreta, las funciones booleanas que definen las divergencias son las siguientes:

$$f(u_1, \ldots, u_m) = \beta, \tag{30}$$

$$f(u_1, \ldots, u_m) = \beta \oplus \bigoplus_{i=1}^{m} u_i u_{i+2}, \tag{31}$$

$$f(u_1, \ldots, u_m) = \beta \oplus \bigoplus_{i=1}^{m} u_i, \tag{32}$$

$$f(u_1, \ldots, u_m) = \beta \oplus \bigoplus_{i=1}^{m} (u_i \oplus u_i u_{i+2}), \tag{33}$$

donde los subíndices se toman módulo m, y $\beta = 0, 1$ para m impar, y $\beta = 0$ para m par.

Se ha demostrado que las clases de divergencias derivadas de las funciones booleanas (30) son no equilibradas y no satisfacen los criterios de propagación. Las clases que se obtienen de las funciones de la forma (31) son equilibradas (con la excepción que se produce cuando m es múltiplo de 4) y satisfacen también el criterio de propagación. Aunque las clases cuya función booleana es (32) son equilibradas, ellas no satisfacen los criterios de propagación. Finalmente, los autómatas celulares definidos por la función booleana (33) son equilibrados para $m = 4, 12, 20, \ldots$ y satisfacen el criterio de propagación.

En consecuencia, los ACE equilibrados cuyas divergencias son también equilibradas son los siguientes: 27, 30, 39, 45,53, 58, 75, 78, 83, 86, 89, 92, 101, 106, 114, 120, 135, 141, 149, 154, 163, 166, 169, 172, 177, 180, 197, 202, 210, 216, 225, 228. Obsérvese que todos ellos pertenecen a las clases II y VII para m impar y a la clase III para m par.

Además, si m no es múltiplo de 4 los ACE cuyas divergencias son equilibradas y satisfacen el criterio de propagación son los siguientes: 1, 4, 11, 14, 19, 22, 25, 28, 32, 37, 42, 47,

50, 55, 56, 61, 67, 70, 73, 76, 81, 84, 91, 94, 98, 103, 104, 109, 112, 117, 122, 127, 128, 133, 138, 143, 146, 151, 152, 157, 161, 164, 171, 174, 179, 182, 185, 188, 194, 199, 200, 205, 208, 213, 218, 223, 227, 230, 233, 236, 241, 244, 251 y 254.

AGRADECIMIENTOS

Este trabajo ha sido subvencionado por el Ministerio de Ciencia e Innovación (España) a través del proyecto MTM2008-02773.

REFERENCES

[1] R. Alonso-Sanz, L. Bull, "Random number generation by cellular automata with memory" en *Internat. J. Mod. Phys. C* vol. 19, no. 2, pp. 351-367, 2008.

[2] P.P. Chaudhuri, D.R. Chowdhury, S. Nandi, S.Chattopadhyay, "Additive Cellular Automata: Theory and Applications, Volume 1", Wiley-IEEE Computer Society Press, Los Alamitos, CA, 1997.

[3] T.W. Cusick ,P. Stănică, "Cryptographic Boolean Functions and Applications", Academic Press, 2009.

[4] J. Escuadra Burrieza, A. Martín del Rey, J.L. Pérez Iglesias, A. Queiruga Dios, G. Rodríguez Sánchez, A. de la Villa Cuenca, "Cryptographic properties of boolean functions defining elementary cellular automata", en *Int. J. Comput. Math.* (en prensa)

[5] A. Martín del Rey, G. Rodríguez Sánchez, "Boolean Differential Operators", en *Inform. Comput.* (Enviado).

[6] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, J. Vandevalle, "Propagation characteristic of boolean functions", en Advances in Cryptology, Proc. of Eurocrypt'90, *Lect. Notes Comput. Sci.* vol. 473, pp. 161-173, 1991.

[7] A. Fúster Sabater, "Cellular Automata in Stream Ciphers", en *Commun. Contemp. Math.*, vol. 477, no. 1, pp. 1-20, 2009.

[8] P. Sarkar, "A brief history of cellular automata", en *ACM Comput. Surv.*, vol. 32, no. 1, pp. 80-107, 2000.

[9] A.F. Webster, S.E. Tavares, "On the design of S-boxes", en Advances in Cryptology, Proc. of Crypto'85, *Lect. Notes Comput. Sci.*, vol. 219, pp. 523-534, 1985.

[10] S. Wolfram, "Cellular Automata and Complexity: Collected Papers", Addison-Wesley, 1994.

[11] S. Wolfram, "A New Kind of Science", Wolfram Media Inc., Champaign, IL, 2002.

Nuevo generador pseudoaleatorio caótico

A. B. Orúe, G. Álvarez, A. Guerra, G. Pastor, M. Romera y F. Montoya
Instituto de Física Aplicada, CSIC
Email: fausto@iec.csic.es

***Abstract*—Se describe una familia de generadores pseudoaleatorios criptográficamente seguros basados en la combinación unidireccional de dos o más secuencias, generadas mediante mapas caóticos lineales a trozos con coeficientes variables dinámicamente y rotación de bits variable dinámicamente. Se describen los conceptos y principios empleados en el diseño de los mismos.**

I. Introducción

Los generadores pseudoaleatorios son fundamentales en criptología ya que la seguridad de muchos sistemas criptográficos depende de la generación de secuencias pseudoaleatorias bien sea en la criptografía de clave simétrica para cifrado en flujo [1], o en la criptografía de clave asimétrica para la generación de claves, vectores de inicialización o números primos. Sin embargo, generar una buena secuencia de números pseudoaleatorios no es una tarea fácil y sigue siendo un tema importante de investigación en ciencias de la computación y criptografía.

El diseño de generadores pseudoaleatorios fiables sigue siendo un punto crítico en criptología. Algunos estándares *de facto* que se creían seguros han fracasado [2], [3], otros generadores que sí son seguros resultan poco útiles por su lentitud, como el BBS [4]. En 2000 comenzó el proyecto europeo NESSIE (*New European Schemes for Signatures, Integrity and Encryption*) que convocó un concurso, para el diseño de primitivas criptográficas. Lamentablemente, los seis cifradores en flujo presentados fallaron frente al criptoanálisis. En 2004 el proyecto europeo eSTREAM convocó el concurso para seleccionar un estándar de cifrador en flujo. Como resultado, en 2008 se seleccionaron siete finalistas del concurso (4 en *software* y 3 en *hardware*) pero actualmente aún no se ha podido decidir cuál de ellos merece ser un estándar.

El vínculo entre la criptografía y los sistemas caóticos está siendo objeto de un intenso estudio. Muchos investigadores están de acuerdo en que la interacción de estas áreas puede ser mutuamente beneficiosa. Muchas herramientas de análisis de los sistemas caóticos han servido igualmente como herramientas en el criptoanálisis de muchos sistemas y para el estudio y perfeccionamiento del diseño de otros [5], [6], [7].

En este trabajo se presenta una familia de generadores pseudoaleatorios que servirán como secuencia cifrante en un cifrador en flujo, partiendo de una aplicación caótica. Para evaluar la aleatoriedad de la secuencia pseudoaleatoria, su salida se compara con la salida de una secuencia realmente aleatoria, mediante pruebas estadísticas.

II. Familia de generadores pseudoaleatorios basados en la combinación de mapas caóticos

La familia de generadores pseudoaleatorios propuesta se basa en la combinación de las secuencias generadas por varios generadores pseudoaleatorios básicos, mediante una función unidireccional. Cada uno de ellos está constituido por un mapa caótico, criptográficamente seguro por sí mismo, que tiene una entropía elevada.

La combinación de varias secuencias mediante la función unidireccional tiene dos objetivos. El primero es conseguir aumentar el número de estados del sistema, con el consiguiente aumento del periodo de repetición, de la entropía y del número de claves. El segundo objetivo es el aumento de la seguridad. En efecto, al mezclar varias secuencias de forma que el tamaño de la palabra de salida sea menor que la suma de los tamaños de las palabras de entrada, resulta imposible hacer un análisis individualizado de las secuencias generadas por cada mapa caótico, dificultando en extremo un ataque criptoanalítico.

Cada mapa tiene un número limitado de estados —y por tanto su período de repetición también es limitado— en función de la longitud de palabra del lenguaje con que se programe, que a su vez depende de la longitud de palabra del hardware que se utilice. El método de combinación elegido consiste en la suma módulo 2 bit a bit de los números generados por cada mapa caótico. El número de mapas debe elegirse en dependencia de la aplicación esperada y del número de bits que tengan las palabras del software con que se programe; dos configuraciones típicas equivalentes serían la combinación de dos mapas caóticos programados con 64 bits o de cuatro mapas caóticos programados con 32 bits.

Se ha utilizado la mezcla de operaciones aritméticas y operaciones orientadas a bit, porque ello sirve para evitar los ataques puramente algebraicos y los ataques puramente orientados a bits. El uso de la mezcla de una variedad de dominios en una forma no lineal y no algebraica, dificulta en extremo la posibilidad de modelizar el comportamiento matemático del esquema.

Los procesadores modernos —cuando se opera con números de igual cantidad de bits que el tamaño de palabra de la máquina— pueden hacer de forma muy eficaz operaciones aritméticas módulo el tamaño de la palabra de la máquina, operaciones booleanas bit a bit y desplazamientos de bits. Todas estas operaciones se combinan en el generador propuesto, logrando así una gran complejidad matemática junto con una elevada eficiencia computacional.

A. Mapa caótico básico

El mapa caótico utilizado $F(x_t, a, c, r)$ es de tipo unidimensional, consistente en la modificación de un mapa lineal a trozos mediante el incremento dinámico de coeficientes y la rotación dinámica de los bits de las muestras, que denominaremos abreviadamente como MLT-CDRD; siendo $\{x_t\}$ la órbita caótica y a_t, c_t, r_t los parámetros de control del sistema, definido por:

$$\begin{aligned} x_t &= ((a_t\, x_{t-1} + c_t) \mod m) \ggg r_t, \\ a_t &= (a_{t-1} + \Delta a) \mod m, \\ c_t &= (c_{t-1} + \Delta c) \mod m, \\ r_t &= (r_{t-1} + \Delta r) \mod n, \end{aligned} \tag{1}$$

donde t es el tiempo; n es el número de bits de precisión utilizado; $m = 2^n$ es el número máximo de valores que puede tomar x_t; a_t, c_t, r_t son coeficientes cuyos valores sufren respectivamente un incremento $\Delta a, \Delta c, \Delta r$ a cada iteración del sistema; siendo x_0, a_0, c_0, r_0 los valores iniciales. El operador $\ggg r$, significa un desplazamiento circular a derechas de r bits de la expresión binaria de la palabra afectada, al que denominaremos abreviadamente *rotación*.

La novedad de este mapa caótico radica en que se realizan dos operaciones de diferente índole concatenadas, la primera es una operación lineal $x_t = (a_t\, x_{t-1} + c_t) \mod m$, propia de los generadores aleatorios algebraicos y la segunda es una rotación de bits $\ggg r_t$, propia de los generadores con registro de desplazamiento. Tanto los coeficientes a_t y c_t como la magnitud de la rotación se varían dinámicamente con el tiempo.

La función $F(x) = ((a_t\, x_{t-1} + c_t) \mod m) \ggg r_0$, para $r_0 =$ constante, es biyectiva y, por tanto, invertible, con la consiguiente posibilidad de ser atacada criptoanalíticamente.

Pero la misma función con $r = r_0 + \Delta r \mod n$ —siendo Δr el incremento que sufre r a cada iteración del sistema— no es biyectiva, es decir que a cada elemento del conjunto de llegada le corresponden varios elementos del conjunto de partida, haciendo imposible su inversión y análisis por un atacante.

Si se dibuja una gráfica bidimensional de los puntos de la órbita caótica $\{x_t\}$ en función de $\{x_{t-1}\}$, se obtiene el llamado mapa de retorno de la función, que en casos sencillos permite reconstruir el valor de los parámetros de una órbita caótica.

A continuación se presentan cinco ejemplos que ilustran la estructura del mapa de retorno del MLT-CDRD para diferentes supuestos: sin rotación ni variación de coeficientes, con variación dinámica de coeficientes a_t y c_t únicamente, con rotación constante r_0 únicamente, con rotación dinámica y, finalmente, con rotación dinámica y variación dinámica de coeficientes.

1) Supuesto sin rotación ni variación dinámica: Si en el MLT-CDRD se toman los valores $\Delta a = \Delta c = \Delta r = r_0 = 0$, el mapa de retorno de la función $x_t = ((a_t\, x_{t-1} + c_t) \mod m) \ggg r_t$, se reduce a un diente de sierra, con tantos tramos como el valor de a_0 y con una ordenada en el origen igual a c_0. En la Fig. 1 se representa este caso para $a_0 = 5$ y $c_0 = 1$, con una precisión $n = 16$ bits. El periodo máximo p

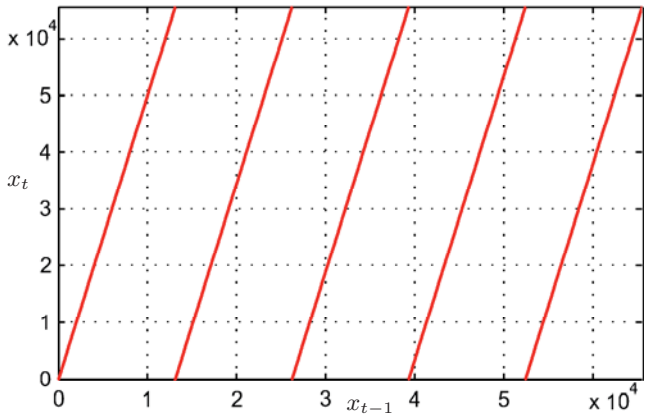


Fig. 1. Mapa de retorno de la función $x_t = (5x_{t-1} + 1) \mod m$, para $m = 2^{16}$.

se alcanza para $a_0 \mod 4 = c_0 \mod 2 = 1$, siendo su valor $p = 2^n = m$, que es el máximo número posible de estados que puede adquirir la función [8].

La función $F(x) = ((a_t\, x_{t-1} + c_t) \mod m) \ggg r_t$, para estos valores de parámetro, es biyectiva y por tanto invertible.

Es evidente que la seguridad proporcionada por un mapa de esta índole es mínima, pues se puede determinar fácilmente el valor de los parámetros que lo controlan por la simple inspección del mapa de retorno, confeccionado a partir de una colección de números generados por él.

Este sencillo mapa coincide con el conocidísimo generador congruencial lineal, cuya falta de seguridad es sobradamente conocida [8], [9].

2) Supuesto con variación dinámica de los coeficientes: Un paso inicial hacia la seguridad del sistema consiste en incrementar a cada iteración los valores de los coeficientes a_t y c_t.

El mapa de retorno correspondiente se presenta en la Fig. 2, para $a_0 = 5$, $\Delta a = 4$, $c_0 = 1$, $\Delta c = 4$, $r_0 = \Delta r = 0$ y $n = 16$ bits; el periodo máximo p se obtiene para $a_t \mod 4 = c_t \mod 4 = 1$, siendo su valor $p = 2^n = m$.

Obsérvese que el mapa de retorno no proporciona ninguna información acerca del valor de los parámetros, ni de su incremento. Pero, como el proceso es lineal, su carácter deter-

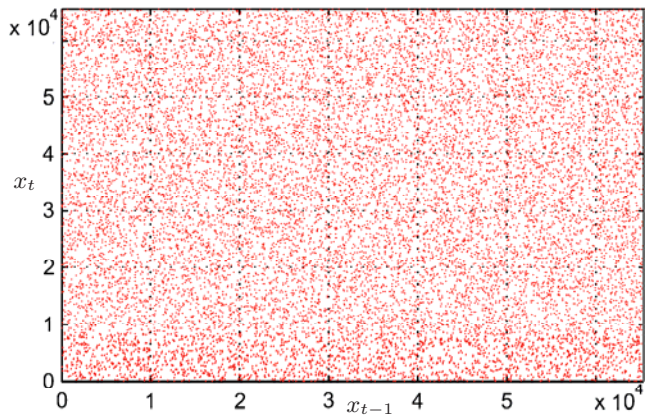


Fig. 2. Mapa de retorno de la función $x_t = (a_t\, x_{t-1} + c_t) \mod m$; para $a_0 = 5$, $a_t = a_{t-1} + 4$, $c_0 = 1$, $c_t = c_{t-1} + 4$, $m = 2^{16}$.

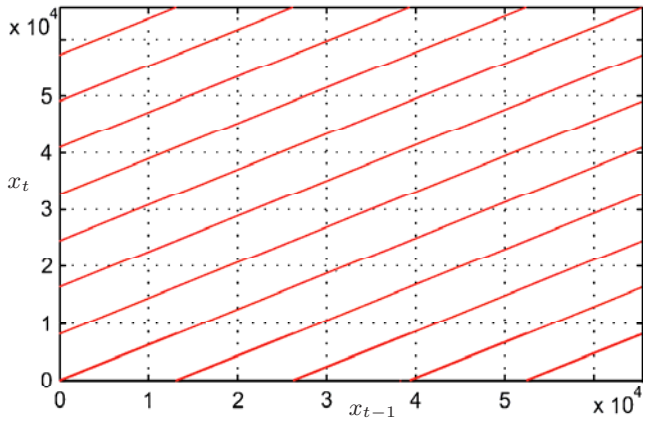


Fig. 3. Mapa de retorno la función $x_t = ((5x_{t-1}+1) \mod m) \ggg r$ para $r = 3$ y $m = 2^{16}$.

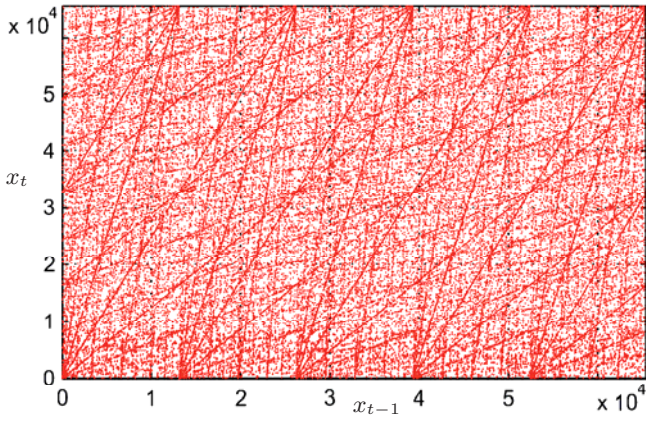


Fig. 4. Mapa de retorno la función $x_t = ((5x_{t-1}+1) \mod m) \ggg r_t$ para $r_0 = 0$, $r_t = r_{t-1}+1$ y $m = 2^{16}$.

minista y la función $x_t = ((a_t\, x_{t-1}+c_t) \mod m) \ggg r_t$, para estos valores de parámetro, es biyectiva y por tanto invertible, siempre será posible calcular el valor de los parámetros a partir del estudio de los sucesivos valores de x_t, resolviendo un sencillo sistema de ecuaciones lineales. Por ello la mejora de la seguridad es limitada.

3) Supuesto con rotación constante: Una alternativa diferente consiste en aplicar una rotación al mapa lineal a trozos. La rotación es una operación que se hace de forma eficiente tanto en software como en hardware, pero que es no lineal y resulta compleja de modelizar algebraicamente, así se tiene que su expresión algebraica es:

$$x \ggg r = \lfloor x/2^r \rfloor + (x\, 2^{n-r}) \mod 2^n.$$

En la Fig. 3 se ilustra el mapa de retorno de la función para los valores de parámetro $\Delta a = \Delta c = \Delta r = 0$, $a_0 = 5$, $c_0 = 1$ y $n = 16$, valores que sólo difieren de los del supuesto primero en que ahora hay una rotación fija $r_0 = 3$.

Se puede comprobar que cada uno de los tramos a_0 del diente de sierra de la Fig. 1 se ha desdoblado en 2^r tramos paralelos con una pendiente 2^r veces menor; resultando aún posible deducir el valor de los parámetros mediante la inspección del mapa de retorno.

La función $F(x) = ((a_t\, x_{t-1} + c_t) \mod m) \ggg r_t$, sigue siendo biyectiva; pero la novedad importante es que el periodo de repetición de la secuencia generada es imprevisible y depende de los valores de los parámetros y de la rotación. Experimentalmente se han encontrado periodos comprendidos entre el 1% y el 80% del periodo sin rotación $p = 2^n$. El efecto es que se ha producido un avance hacia la impredecibilidad de la secuencia, al precio de una reducción del periodo.

4) Supuesto con rotación dinámica: Otra mejora de la entropía consiste en modificar el caso anterior variando cíclicamente el valor de la rotación.

En la Fig. 4 se ilustra un caso igual al anterior con la sola diferencia de que ahora se hace $r_0 = 0$ y $\Delta r = 1$. Se puede observar que el mapa de retorno es extremadamente complejo, aunque aún se puede intentar alguna estimación del valor de los parámetros.

El resultado importante es que no sólo ha aumentado la confusión del mapa de retorno, sino también el periodo de repetición de la secuencia, se observan ocurrencias de periodos comprendidos entre el 20% y el 1600% del periodo sin rotación $p = 2^n$. Los periodos más cortos se originan para los valores de Δr que tienen más factores primos comunes con n.

La función $F(x) = ((a_t\, x_{t-1} + c_t) \mod m) \ggg r_t$, para $\Delta r \neq 0$ no es inyectiva ni sobreyectiva, ya que a cada elemento del conjunto de llegada le puede corresponder más de un elemento del conjunto de partida.

5) Supuesto con rotación dinámica y variación dinámica de coeficientes: Finalmente, en la versión completa del MLT-CDRD se consigue la entropía óptima combinando todos los mecanismos anteriormente descritos.

La Fig. 5 ilustra el mapa de retorno para los valores de parámetro: $a_0 = 5$, $c_0 = 1$, $r_0 = 0$, $\Delta a = 4 \times 3$, $\Delta c = 4 \times 1$, $\Delta r = 3$ y $n = 16$. Nótese que el mapa de retorno contiene muchos más puntos que cualquiera de los sistemas anteriores, debido a que el período es mucho más largo y a una mayor entropía.

Los valores de los coeficientes han de cumplir las siguientes condiciones para maximizar el periodo de repetición: $a_0 \mod 4 = c_0 \mod 2 = 1$, $\Delta a \mod 4 = \Delta c \mod 2 = 0$, tal como se demuestra en [8]§3.2.1.2, Teo. A, cuando se estudia el generador lineal congruencial. Nótese que estas condiciones han de cumplirse aunque el valor de los parámetros varíe dinámicamente.

Los valores r_0 y Δr han de elegirse de tal forma que se maximice el periodo de repetición de la sucesión de los valores de la rotación, para conseguir el máximo de entropía; lo que conduce a que Δr sea relativamente primo con n; r_0 puede ser cualquier valor $0 \leq r_0 < n$.

En la Fig. 5 se ha representado un número de puntos $N = 4m = 2^{18}$ de la órbita del generador. Naturalmente, si se siguiesen representando más puntos se rellenaría toda la superficie del mapa de retorno, es decir que el mapa es completamente uniforme. Este hecho no ocurría en ninguno de los supuestos anteriores ya que cuando los puntos se agrupaban en rectas era evidente que la uniformidad era inexistente. En el caso (aparentemente mejor) de la Fig. 2, la uniformidad es limitada ya que si se dibujase más de un periodo los puntos de los periodos sucesivos caerían sobre los dibujados

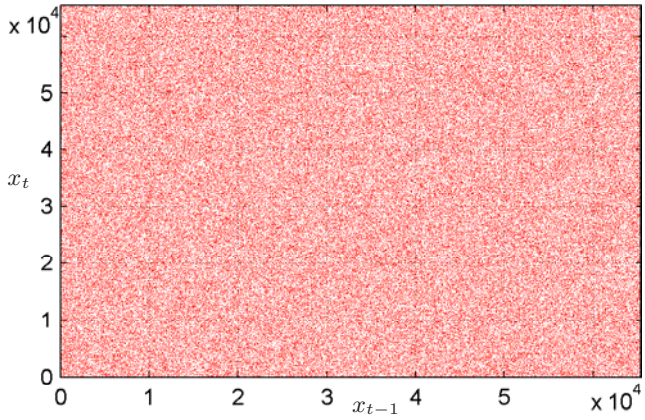


Fig. 5. Mapa de retorno la función $x_t = (a_t x_{t-1} + c_t \mod m) \ggg r_t$ para $a_0 = 5$, $a_t = a_{t-1} + 12$, $c_0 = 1$, $c_t = c_{t-1} + 4$, $r_0 = 5$, $r_t = r_{t-1} + 3$ y $m = 2^{16}$.

anteriormente (el periodo de la órbita era $p = m = 2^{16}$).

En la versión completa con $n = 16$ bits hemos observado ocurrencias de periodos comprendidos entre 132 y 66000 veces la longitud del periodo sin rotación $p = 2^n$, para diferentes valores de los parámetros, manteniéndose estas proporciones cuando se codifica con mayor cantidad de bits. Es decir, que se consiguen períodos notablemente más largos que los esperables con el sencillo generador del supuesto primero.

La función $F(x) = ((a_t\, x_{t-1} + c_t) \mod m) \ggg r_t$, para $\Delta r \neq 0$ no es inyectiva ni sobreyectiva ya que a cada elemento del conjunto de llegada le puede corresponder más de un elemento del conjunto de partida. El número M de elementos diferentes entre sí del conjunto de llegada obedece a la fórmula del problema del cumpleaños:

$$M = m \left(\frac{m-1}{m} \right)^{N-1}, \qquad (2)$$

siendo N el número de muestras generadas. Para una precisión $n = 16$ bits, el número máximo de muestras diferentes es $m = 2^{16}$. Si se prueban exactamente los m valores posibles del conjunto de partida, sólo aparece un $63,22\%$ de valores diferentes del conjunto de llegada, es decir el $36,78\%$ de los elementos de llegada están repetidos. Por tanto, no es posible invertir la función, ya que aproximadamente un tercio de los elementos del conjunto de llegada tiene más de una preimagen.

Las muestras generadas por el mapa $x_t = ((a_t\, x_{t-1} + c_t) \mod m) \ggg r_t$, también están de acuerdo con la Ec. 2; es decir, para un número de muestras generadas igual a $N = 2^{16}$, las muestras diferentes son $M = 0,63\, N$, para $N = 2^{17}$ las muestras diferentes son $M = 0,86\, N$, para $N = 2^{18}$ las muestras diferentes son $M = 0,98\, N$ y para para $N = 2^{19}$ las muestras diferentes son $M = N$.

Este comportamiento estadístico es exactamente igual al que se podría esperar de una fuente de números totalmente aleatoria, lo que confirma la perfecta pseudoaleatoriedad de la secuencia generada.

Las secuencias de números generados por el MLT-CDRD, (en su versión con $n = 64$ bits) pasan con éxito todas las pruebas de la batería de tests de aleatoriedad del American National Institute of Standards and Technology, recogidos en la NIST SP 800-22 [10], así como las más exigentes *Diehard* de Marsaglia de 1995 y también los nuevos, y aún más exigentes, Tuftests de Marsaglia y Tsang de 2002 [11].

La versión completa del MLT-CDRD programada en C99 —estándar ISO/IEC 9899:TC3— con números enteros de precisión extendida de 64 bits significativos, ha conseguido un rendimiento de $2,5$ bits por ciclo de reloj en un ordenador de tipo PC con procesador Intel.

Una interesante característica de este mapa caótico, es que puede generar simultáneamente múltiples secuencias independientes, todas de idéntico periodo, con un esfuerzo computacional adicional mínimo, simplemente haciendo simultáneamente varias operaciones de rotación con parámetros r_0 y Δr de valores diferentes.

B. Coeficiente de Lyapunov

Para que exista caos en un sistema determinístico debe haber dependencia sensible a las condiciones iniciales. Ello significa que las órbitas de dos puntos iniciales próximos divergerán exponencialmente. Siguiendo a Moon [12], si d_0 es una medida de la distancia inicial entre dos órbitas, y d_k es una medida de la distancia entre las mismas órbitas al cabo de k iteraciones, se define,

$$\frac{d_k}{d_0} = 2^{\Lambda k}, \quad \Lambda = \frac{1}{k} \log_{10} \frac{d_k}{d_0},$$

siendo Λ el exponente de Lyapunov correspondiente a k iteraciones. La base sobre la que se aplica el exponente puede elegirse según convenga. Se ha tomado la base 10 porque se trabaja con números enteros.

El exponente de Lyapunov sirve para medir el caos de un sistema. Si $\Lambda = 0$, la distancia se mantiene y no se puede afirmar que exista caos. Si $\Lambda < 0$, la distancia disminuye, el sistema converge y no es en absoluto caótico. Si $\Lambda > 0$, la distancia aumenta, hay dependencia sensible a las condiciones iniciales, se produce una divergencia exponencial de la órbita y ésta es tanto más caótica cuanto mayor sea el valor de Λ.

En la mayoría de los problemas físicos la órbita está acotada y, por tanto, d_k no tiende a infinito al aumentar el número de iteraciones. La divergencia de las órbitas debe entenderse sólo como localmente exponencial; por ello, se han calculado sucesivamente los exponentes de Lyapunov cada n iteraciones (que sería el periodo de las rotaciones) y se ha tomado la media al cabo de un número K de iteraciones suficientemente grande

$$\Lambda = \lim_{K \to \infty} \frac{1}{K} \sum_{k=1}^{K} \log_{10} \frac{d_{n_k}}{d_{0_k}},$$

donde d_{0_k} y d_{n_k} son las distancias inicial y final en la iteración k.

En la Fig. 6 se ilustra la variación del exponente de Lyapunov en función los valores posibles de a_0, para el mismo ejemplo del supuesto quinto (para su cálculo se ha tomado $K = n$). La media de los valores de los coeficientes de Lyapunov alcanza el valor $0,2605$, para todos los valores de a_0 que cumplen, $a_0 \mod 4 = 1$. Nótese que todos los exponentes están agrupados en una estrecha franja de valores,

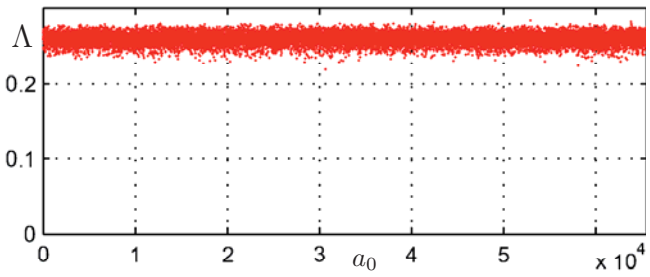


Fig. 6. Exponentes de Lyapunov en función de a_0 de la versión completa del MLT-CDRD.

garantizando la rápida divergencia de las órbitas para todos los a_0.

C. Combinación de los mapas

La versión completa del mapa MLT-CDRD puede ser por sí misma un excelente generador de números pseudoaleatorios; pero como en cualquier diseño de esta índole quedan algunos cabos sueltos que requieren ser totalmente asegurados.

El primer problema consiste en que si el MLT-CDRD se programase con palabras de pocos bits, por ejemplo con $n = 32$, el periodo de repetición podría resultar relativamente pequeño para ciertas aplicaciones que requiriesen períodos extremadamente largos.

El segundo problema radica en que nadie puede garantizar que no aparezcan puntos fijos o ciclos cortos del generador para algún conjunto de parámetros y valores iniciales. El carácter totalmente impredecible del generador, debido a su estructura no lineal —que mezcla operaciones algebraicas y operaciones bit a bit— solamente permite hacer estimaciones estadísticas de su comportamiento. La experiencia demuestra que la posibilidad de puntos fijos o ciclos cortos es muy remota, pues se han realizado pruebas exhaustivas del comportamiento para valores pequeños de n (concretamente para 6 bits) y numerosos ensayos para $n = 32$ y $n = 64$ bits, todos ellos totalmente satisfactorios en cuanto al período de repetición y a la aleatoriedad.

El tercer problema es que en el MLT-CDRD los números generados que aparecen a la salida son los mismos que se emplean como entrada para generar la muestra futura, permitiendo a un atacante comparar la salida y entrada del mapa, lo que posibilitaría intentar algún tipo de análisis dirigido a estimar matemáticamente el valor de los parámetros o parte de ellos.

Para hacer frente a todos estos inconvenientes se propone una familia de generadores basados en la combinación de dos o más MLT-CDRD mediante una función unidireccional.

A continuación se describe el ejemplo más sencillo posible, en el que se combinan dos mapas caóticos programados con palabras de 64 bits, como se ilustra en la Fig. 7, que mezcla dos secuencias $\{x_t\}$ e $\{y_t\}$ mediante la suma XOR bit a bit, de forma que se resuelven satisfactoriamente todos los problemas anteriores.

El primer mapa está definido por las Ecs. 1 y el segundo

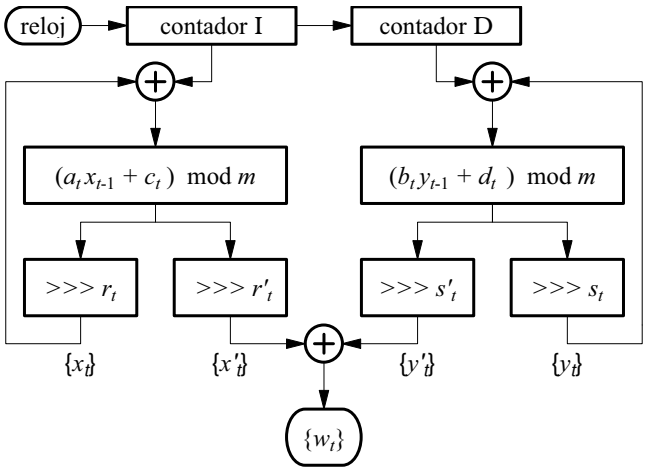


Fig. 7. Generador pseudoaleatorio completo, combinando dos MLT-CDRD.

es:

$$\begin{aligned} y_t &= ((b_t\, y_{t-1} + d_t) \mod m) \ggg s_t, \\ b_t &= (b_{t-1} + \Delta b) \mod m, \\ d_t &= (d_{t-1} + \Delta d) \mod m, \\ s_t &= (s_{t-1} + \Delta s) \mod n, \end{aligned} \tag{3}$$

siendo $a_0 \neq b_0$, $c_0 \neq d_0$, $\Delta a \neq \Delta b$, $\Delta c \neq \Delta d$; $r_0 \neq s_0$, $\Delta r \neq \Delta s$, para asegurar que las secuencias $\{x_t\}$ y $\{y_t\}$ sean absolutamente diferentes.

Igualmente se podrían combinar mediante la suma XOR bit a bit tantos MLT-CDRD como fuesen necesarios para alcanzar el periodo de repetición deseado.

El primer problema queda resuelto por la combinación de varios mapas MLT-CDRD, por lo que el número de estados del generador se amplia tremendamente. Si se estaba trabajando con palabras de 64 bits, el número de estados pasa de $2^{64} \simeq 1,8 \times 10^{19}$ a $2^{128} \simeq 3,4 \times 10^{38}$.

El segundo problema se resuelve con la perturbación de la entrada de cada mapa x_t e y_t mediante la suma XOR bit a bit de las salidas de cada uno, x_{t-1} e y_{t-1}, con el estado de dos contadores (I y D) de igual numero de bits cada uno que el MLT-CDRD. Ambos contadores están en cascada; es decir, cada vez que el contador I da una vuelta completa se incrementa en una cuenta el contador D. De esta forma se garantiza que el periodo sea, como mínimo, el periodo de los dos contadores en cascada. Si se trabaja con palabras de 64 bits, el número de estados garantizado sería $2^{128} \simeq 3,4 \times 10^{38}$.

El tercer problema queda resuelto por la mezcla de varias secuencias en una sola, de forma que el tamaño de la palabra de la secuencia combinada $\{w_t\}$ sea igual al tamaño de las palabras de cada una de las secuencias $\{x_t\}$ e $\{y_t\}$ a combinar, resultando imposible hacer un análisis individualizado de las secuencias generadas por cada mapa caótico. Para potenciar al máximo esta operación de ocultación, no se suman XOR bit a bit directamente las secuencias $\{x_t\}$ e $\{y_t\}$, sino unas variantes de estas $\{x'_t\}$ e $\{y'_t\}$, que difieren de las anteriores en que se utilizan rotaciones diferentes $r'_t \neq r_t$ y $s'_t \neq s_t$, para generarlas.

La clave del generador está constituida por los coeficientes

de los mapas caóticos y las semillas x_0 e y_0 y los valores iniciales de los contadores pueden también formar parte de la clave o bien ser conocidos. Debido a las limitaciones impuestas en el supuesto quinto, el número de bits con que se codifican los coeficientes es: $n-2$ bits para $a_0, b_0, \Delta a, \Delta b$; $n-1$ bits para $c_0, d_0, \Delta c, \Delta d$; $(\log_2 n)-1$ bits para Δr y Δs; y $\log_2 n$ bits para r_0 y s_0. Si se trabaja con palabras de 64 bits, el número total de bits de clave sería $4\times 62 + 4\times 63 + 2\times 5 + 2\times 6 = 522$. Ahora bien, dado que los coeficientes utilizados por un MLT-CDRD no deberían repetirse en el otro MLT-CDRD (para generar secuencias radicálmente diferentes en cada uno), se puede considerar que el número total efectivo de bits de los coeficientes ha de reducirse a unos 521. Si se suman los bits de las semillas (haciendo conocido los valores iniciales de los contadores) el número final de bits de la clave sería 649 y el número final de claves 2^{649}.

1) Seguridad: La forma evidente de atacar el sistema, cuando se conoce la secuencia de números de salida y el estado de los contadores, es la prueba exhaustiva de claves; pero dada su cantidad la operación es prohibitiva. Existe la posibilidad de un ataque por encuentro a medio camino; pero ello exigiría generar y almacenar un promedio de 2^{325} conjuntos de al menos 10 muestras de salida, lo que también es inviable.

2) Aleatoriedad: Las secuencias de números generados por el generador combinado también pasan con éxito todas las pruebas de la batería de tests de aleatoriedad SP 800-22 [10], las *Diehard* de Marsaglia así como los Tuftests de Marsaglia y Tsang [11].

3) Velocidad: El precio pagado por la combinación de dos MLT-CDRD —igualmente programada en C99 con enteros con precisión extendida de 64 bits significativos—, ha sido que el rendimiento se ha reducido a 1 bit por ciclo de reloj en un ordenador de tipo PC con procesador Intel y SO Windows32. Hay que señalar que esta velocidad está al nivel de la conseguida por los finalistas del concurso eSTREAM.

III. Aplicaciones

La familia de generadores descrita es apropiada para usarse en aplicaciones criptográficas especialmente exigentes, como la generación de claves y el cifrado en flujo.

Dentro del proyecto CENIT SEGUR@ encargado por Telefónica I+D al CSIC, se han realizado varios desarrollos basados en este generador, entre ellos un cifrador en flujo síncrono, un cifrador en flujo autosincronizante, un generador de claves y un generador de MAC's de archivos. Estos desarrollos se han aplicado en telefonía móvil al cifrado de SMS para móviles con SO Windows Mobile $6.0/6.5$ y Symbian y para cifrado de correo electrónico y archivos en Windows Mobile $6.0/6.5$ en móviles 3G; funcionando satisfactoriamente en tiempo real.

Se ha solicitado la concesión de una patente internacional a nombre de Telefónica S.A.

IV. Conclusión

Se ha descrito el diseño de una familia de generadores pseudoaleatorios criptográficamente seguros basados en la combinación unidireccional de dos o más secuencias, generadas mediante mapas caóticos lineales a trozos con coeficientes variables dinámicamente y rotación de bits variable dinámicamente. El generador es impredecible, de periodo de repetición mínimo garantizado, y satisface los test de aleatoriedad actualmente más exigentes. La velocidad de generación alcanzada es de un bit por ciclo de reloj en un PC con procesador Intel y SO Windows32.

La característica destacada es la combinación de operaciones algebraicas con operaciones booleanas y desplazamientos circulares, cuya asociación maximiza la seguridad y la entropía del sistema.

Agradecimientos

Los autores agradecen su ayuda al Ministerio de Educación y Ciencia proyectos CUCO MTM(2008-02194) y TEC2009-13964-C04-02; al CDTI (Ministerio de Industria, Turismo y Comercio) en colaboración con Telefónica I+D, proyecto SEGUR@ (CENIT 2007-2010).

Referencias

[1] G. S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *J. Am. Inst. Elec.Eng.*, vol. 45, pp. 109–115, 1926.

[2] A. Klein, "Attacks on the RC4 stream cipher," *Designs, Codes and Cryptography*, vol. 48, no. 3, pp. 269–286, 2008.

[3] I. Goldberg and I. Wagner, "Randomness and the netscape browser," *Dr. Dobb's Journal*, pp. 66–70, 1996.

[4] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," *SIAM Journal on Computing*, vol. 15, pp. 364–383, 1986.

[5] G. Alvarez, F. Montoya, M. Romera, and G. Pastor, "Cryptanalysis of a chaotic encryption system," *Phys. Lett. A*, vol. 276, no. 1-4, pp. 191–196, 2000.

[6] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129–2151, August 2006.

[7] A. B. Orue, V. Fernandez, G. Alvarez, G. Pastor, , M. Romera, S. Li, and F. Montoya, "Determination of the parameters for a Lorenz system and application to break the security of two-channel chaotic cryptosystems," *Phys. Lett. A*, vol. 372, pp. 5588–5592, 2008.

[8] D. E. Knuth, *The Art of Computer Programming*, 3rd ed. Addison-Wesley, 1997, vol. 2, Seminumerical Algorithms.

[9] M. Bellare, S. Goldwasser, and D. Micciancio, "Pseudo-random number generation within cryptographic algorithms: The dds case," in *Advances in Cryptology - CRYPT0 '97*, B. Kaliski, Ed. Springer Verlag, 1997, vol. LNCS 1294, pp. 277–291.

[10] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, revision-1 ed., National Institute of Standards and Technology (NIST), U.S. Department of Commerce, August 2008.

[11] G. Marsaglia and W. W. Tsang, "Some difficult-to-pass tests of randomness," *Journal of Statistical Software*, vol. 7, no. 3, pp. 1–9, Jan 2002.

[12] F. C. Moon, *Chaotic and fractal dynamics*, Jhon Wiley and Sons, 1992.

On the inadequacy of unimodal maps for cryptographic applications

David Arroyo * § , José María Amigó†, Shujun Li‡ and Gonzalo Alvarez*
*Instituto de Física Aplicada, Consejo Superior de Investigaciones Científicas
Email: {david.arroyo, gonzalo}@iec.csic.es
†Centro de Investigación Operativa, Universidad Miguel Hernández
Email: jm.amigo@umh.es
‡Department of Computer and Information Science, University of Konstanz
Web site: www.hooklee.com
§Present address: Instituto de Acústica, Consejo Superior de Investigaciones Científicas, Serrano 144, 28006 Madrid

***Abstract*—The security of chaos-based cryptosystems is closely related to the possibility of recovering control parameters and/or initial conditions from partial information on the associated chaotic orbits. In this paper we analyze this possibility for the case of unimodal maps. We show a meaningful set of contexts where the dynamics of unimodal maps can be reconstructed, which means a relevant reduction of the scope where this kind of chaotic maps can be applied to build up new encryption procedures.**

I. Introduction

Chaos-based cryptography uses chaotic systems to guide the encryption procedure inside an encryption architecture. The examination of the adequacy of a specific chaotic map for an encryption architecture is a very complex problem, and we have the feeling that a general solution to this problem cannot be found. As a matter of fact, even for non-chaos-based encryption systems, it is not possible to establish a general security evaluation procedure. Therefore, the evaluation of the security of a cryptosystem is generally an *ad hoc* procedure, and the starting point should be the set of strategies used by cryptanalysts. The first step in either the design or the security analysis of a cryptosystem is to detect the components which could be examined or studied using previous cryptanalysis techniques. In other words, it is necessary to identify the critical components of a cryptosystem before starting its design or cryptanalysis.

In the case of chaos-based cryptosystems, the identification of these critical components must focus in a first approximation on three points: the selection of the encryption architecture, the selection of the chaotic system(s) and the procedure that determines the association between the chaotic system(s) and the encryption architecture. With respect to the selection of the encryption architecture, if we assume symmetric cryptography, it is necessary to discern between stream ciphers and block ciphers. In [1, Chapters 3 and 4] a detailed analysis of various attacks on conventional stream and block ciphers can be found. For chaos-based cryptography, it is natural that those attacks should anyway be considered, by considering that now the cryptosystems under study are driven by chaos. Concerning the selection of the chaotic system(s), we have to examine thoroughly two critical aspects: (i) the complexity of the chaotic systems; (ii) the possibility of reconstructing the dynamics of the chaotic system(s) from the information leaked, in different cryptanalysis contexts associated to a given encryption architecture.

The complexity of the underlying chaotic systems depends on both their dimensionality and their physical implementation. According to Poincaré-Bendixson Theorem [2, p. 101], chaotic dynamical systems in continuous time have a phase space of dimension greater than 2. Conversely, dynamical systems in discrete time can be chaotic even when the phase space is of dimension 1, if the rule of evolution is a non-invertible function. On the other hand, chaotic systems can be implemented in analog (i.e., upon some circuitry) or in digital form. The first option is generally associated to the use of chaos synchronization techniques [3], [4], which is not the case in the second option. The digital alternative demands an analytical description of the chaotic system. If the chaotic system is described in continuous time, then its analytical definition is a set of differential equations, and the determination of its temporal evolution requires the use of numerical methods. The use of such methods informs about an extra burden (in terms of computation) when calculating the orbits of the chaotic systems. Moreover, it incorporates an extra problem, since numerical methods are defined in dependence of configuration parameters. These parameters must be selected carefully, otherwise the dynamics of the resulting orbits can be modified resulting in a non-chaotic behavior (this is the case of the cryptosystem that we have analyzed in [5]). Contrariwise, chaotic systems in discrete time are given by a set of difference equations, and their orbits can be derived straightforwardly.

With respect to the security of chaos-based cryptosystems, the synchronization techniques entail some critical problems. The conditions required for the synchronization of different chaotic systems are too demanding and amount to weakening the security requirements of an encryption procedure. Certainly, if synchronization is used as the bearer of an encryption architecture, then the chaotic systems at both sides will work using a subset of the control parameters space. Assuming that the control parameters are the key or part of the key of a

chaos-based cryptosystem, the matching sensitivity leads to a narrowing of the key space, thus lessening the computational complexity of a *brute force attack* [6]–[17]. As a result, chaotic systems in discrete time (also known as chaotic maps) are better choices when designing new encryption procedures, since they possess less computational complexity and can be used to construct cryptosystems without synchronization.

Having as aim the concretion of efficient (and secure) chaos-based cryptosystems, it seems that the best option is to select the simplest chaotic maps. This being the case, the logistic map in particular, and unimodal maps in general have been broadly used in the context of chaos-based cryptography [18]–[36]. Nevertheless, we point out that unimodal maps cannot be applied to cryptography straightforwardly. Indeed, it is necessary to examine their potentiality to build up secure cryptosystems. This analysis is performed through the evaluation of chaotic orbits as the kernel of confusion and diffusion of the encryption procedure. In this regard a quantification of the level of "chaoticity" is required, and we also need to identify those situations enabling the estimation of control parameters and/or initial conditions from observed information about the orbits.

The rest of the paper is organized towards the above-described goals as follows. First, we introduce the basic notations used in the following sections. In Sec. III the potentiality of achieving information diffusion by concealing initial conditions of unimodal maps is studied. The analysis of the potentiality for information diffusion also requires to study the dependency of the orbits on control parameters, which is discussed in Sec. IV. Furthermore, the information confusion property is studied in Sec. V by means of different measures of entropy for unimodal maps. Finally, the ergodicity of unimodal maps is analyzed in Sec. VI, which leads to the final comments and conclusions in the last section.

II. Mathematical definition of the scope under consideration

Since we are mainly interested in families of (unimodal) maps, we define an m-dimensional discrete-time dynamical system as a triple $(\Lambda, \mathcal{U}, f)$, where $\Lambda \subset \mathbb{R}^d$ is the set of parameters, $\mathcal{U} \subset \mathbb{R}^m$ is the state space, and $f : \Lambda \times \mathcal{U} \to \mathcal{U}$ is the map that updates the states $x \in \mathcal{U}$ according to the rule $x \mapsto f(\lambda, x)$. Since the parameter λ is held fixed when studying the dynamical aspects, the notation $f(\lambda, x) \equiv f_\lambda(x)$ will be used. Hence, the rule that transforms an state $x_n \in \mathcal{U}$ into an state $x_{n+1} \in \mathcal{U}$ will be written as the difference equation $x_{n+1} = f_\lambda(x_n)$. Accordingly, the forward orbit generated from an initial condition $x_0 \in \mathcal{U}$ is

$$\gamma^+_{f_\lambda}(x_0) = \left\{ f^{(0)}_\lambda(x_0), f^{(1)}_\lambda(x_0), \ldots, f^{(i)}_\lambda(x_0), \ldots \right\}, \quad (1)$$

where

$$f^{(i)}_\lambda(x_0) = \begin{cases} x_0, & \text{if } i = 0 \\ f_\lambda(f^{(i-1)}_\lambda(x_0)), & \text{if } i > 0 \end{cases} \quad (2)$$

If the map $f_\lambda(x)$ is invertible, then the dynamical system $(\Lambda, \mathcal{U}, f)$ is said to be invertible; in this case, one can also defined the backward orbits in a similar way. In this paper the focus is a specific class of maps, namely, the unimodal maps, which are denoted by $\mathcal{F}$. A map $f_\lambda : \mathcal{U} \to \mathcal{U}$, where $\mathcal{U} = [a, b] \subset \mathbb{R}$, is unimodal if it is continuous, has a single turning point (usually called the critical point) x_c in $\mathcal{U}$, and is monotonically increasing (or decreasing) on the left side of x_c and decreasing (or increasing) on the right side.

Two different situations are considered in this paper:

1) The control parameter determines the maximum value of the map, being the critical point independent of the control parameter. In this case, the parametric function f_λ is given by
$$f_\lambda(x) = \lambda F(x), \quad (3)$$
where $F \in \mathcal{F}$ and $F(x_c) = F_{\max}$. The subclass of maps $f_\lambda \in \mathcal{F}$ complying with this description will be denoted by $\mathcal{F}_1$.
As representatives of the map class $\mathcal{F}_1$ we consider the following three maps: a) the logistic map, defined by the rule of evolution
$$x_{n+1} = f_\lambda(x_n) = \lambda \cdot x_n \cdot (1 - x_n)$$
$$\lambda \in [0, 4],\ \mathcal{U} = [0, 1] \quad (4)$$
b) the Mandelbrot map, given by
$$x_{n+1} = f_\lambda(x_n) = x_n^2 + \lambda,$$
$$\lambda \in [-2, 0.25], \mathcal{U} = [-2, 2] \quad (5)$$
and c) the (symmetric) tent map, whose difference equation is
$$x_{n+1} = f_\lambda(x_n) = \begin{cases} \lambda \cdot x_n, & \text{if } 0 \le x_n < 1/2, \\ \lambda \cdot (1 - x_n), & \text{if } 1/2 \le x_n \le 1, \end{cases} \quad (6)$$
with $\lambda \in [1, 2]$, and $\mathcal{U} = [0, 1]$. Strictly speaking, the Mandelbrot map is not included in the map class $\mathcal{F}$. Nevertheless, the Mandelbrot map is topological conjugate with the logistic map [37, p. 529], which implies their equivalency by means of their dynamics.
2) The critical point is given as a function of the control parameter, i.e., $x_c = f(\lambda)$. This leads to a new subclass of maps $\mathcal{F}_2$.

In this paper we consider the skew (full) tent map as a representative of the map class $\mathcal{F}_2$. This map is defined as
$$x_{n+1} = f_\lambda(x_n) = \begin{cases} x_n/\lambda, & \text{if } 0 \le x_n < \lambda, \\ (1 - x_n)/(1 - \lambda), & \text{if } \lambda \le x_n \le 1, \end{cases} \quad (7)$$
with $\lambda \in (0, 1)$, and $\mathcal{U} = [0, 1]$.

III. Measuring the sensitivity to initial conditions

Chaotic systems are deemed adequate for cryptography due to their high sensitivity to both initial conditions and control parameters. With respect to the initial conditions, this sensitivity can be measured by the Lyapunov Exponent (or LE in short) [38]. In this regard, if a chaotic system is used to implement an encryption scheme, then the value(s) of the control

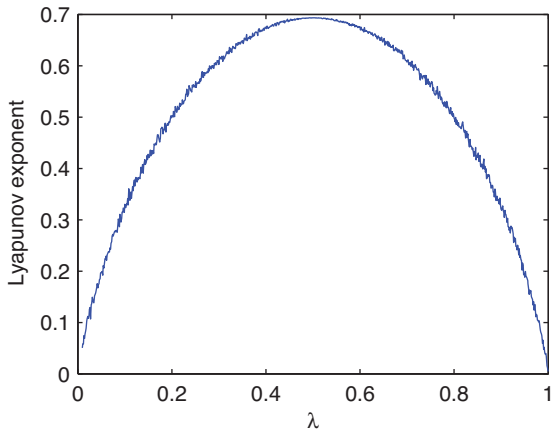


Fig. 1. Lyapunov Exponent of the skew tent map.

parameter(s) must be selected in such a way that the maximum LE is always positive. Quadratic maps possess a dense set of periodic windows in which the maximal LE is not positive [39]. This implies an additional complexity in the selection of adequate values for the control parameter(s). Therefore, it is advisable to use a map with a positive maximum LE for all the values of the control parameters. In this regard, the skew tent map seems to be a good option. Nevertheless, the LE of the skew tent map shows a low value for a large set of values of λ (Fig. 1), which reduces the number of valid methods of the information diffusion process built upon the orbits of the skew tent map. In other words, we should use maps exhibiting robust chaos [40], which can be generated from unimodal maps according to the scheme described in [41]. Finally, we must emphasize that the computation of the LE must be carried out taking into account finite-precision arithmetic, which is the real context of digital chaos-based cryptography. In this sense, the discrete LE [42] should be analyzed and computed.

IV. Study of the sensitivity to control parameter

One characteristic of chaotic systems is that their evolution in time is sensitive to the vector of control parameter(s) λ, i.e., two very close values of λ will eventually lead to very different orbits after a transient number of iterations. Moreover, this difference may be also present when comparing orbits as a whole, i.e, from an statistical point of view. In the context of chaos-based cryptography, it is highly advisable to avoid any kind of dependence of the statistics of the orbits on λ. If some of the statistics of the orbits can be expressed as a function of λ, then an estimation of the control parameters could be performed. For the sake of clarity, the problem is formulated mathematically as follows. Given a chaotic map $f_\lambda : \mathcal{U} \to \mathcal{U}$ and a generating partition $\mathcal{A} = A_0 \cup A_1 \cup \ldots \cup A_{N-1}$, let p_i be the probability of visiting the interval A_i is determined for $0 \leq i \leq N-1$. If the statistical behavior of the map depends on the value of λ, then $p_i = p_i(\lambda)$ and the dependency of p_i with respect to λ can be computed using some kind of statistical distance. Here, we give an example based on the *Wootters' distance* [43]. Let us consider two probability distributions $\boldsymbol{P}_i = \left\{p_j^{(i)}, j = 1, \ldots, N\right\}$ with $i = 1, 2$. The Wootters' statistical distance is given by

$$\mathcal{D}_W(\boldsymbol{P}_1, \boldsymbol{P}_2) = \cos^{-1}\left(\sum_{j=1}^{N} \sqrt{p_j^{(1)} \cdot p_j^{(2)}}\right). \tag{8}$$

If $f_\lambda : \mathcal{U} \to \mathcal{U}$ is unimodal with $\mathcal{U} = [0, 1]$, then an orbit of length M generated from $x_0 \in \mathcal{U}$ can be encoded into a binary sequence,

$$\begin{aligned}&\mathbf{B}_M(f_\lambda, x_0) = \{B_i(f_\lambda, x_0)\}_{i=0}^{M-1} = \\ &= \theta(f_\lambda^{(0)}(x_0))\theta(f_\lambda^{1}(x_0)) \ldots \theta(f_\lambda^{(M-1)}(x_0)),\end{aligned}$$

where $\theta(\cdot)$ is the step function

$$\theta(y) = \begin{cases} 0, & \text{if } y < x_c, \\ 1, & \text{if } y \geq x_c. \end{cases} \tag{9}$$

A probability distribution can be obtained from $\mathbf{B}_M(f_\lambda, x_0)$ by just grouping all bits in a sliding window of length w. As a result, a binary sequence of length M is transformed into a sequence of $M - w + 1$ w-bit integers (or words). The probability distribution associated to $\mathbf{B}_M(f_\lambda, x_0)$ is determined by counting the number of occurrences of each word and dividing the result by $(M - w - 1)$. Wootter's distance can be used, for example, to estimate the control parameter of the tent map. This task is carried out by computing Wootter's distance from the binary sequence $\mathbf{B}_M(f_{\hat{\lambda}}, x_0)$ (generated with an unknown value $\hat{\lambda}$ of the control parameter) to the binary sequences generated with λ ranging in an interval. These distances are computed in Fig. 2 for two values of $\hat{\lambda}$ with $M = 10^4$ and $w = 10$; the corresponding binary sequences were generated with different initial conditions. Figure 2 shows that around the right value of λ there exists a basin of attraction, which leads immediately to an estimation of $\hat{\lambda}$.

Wootters' distance can also be used to distinguish the binary sequences of a unimodal map from those corresponding to another unimodal map [44]. As a matter of fact, this is a relevant application of statistical distances in the context of unimodal maps, since the estimation of the control parameter and the initial condition can be performed from binary sequences without any auxiliary tool but the theory of symbolic dynamics [45]–[48]. In addition, we must take into account that this method is feasible only when the two maps involved are not topologically conjugate [37, p. 529].

V. Analysis of chaotic orbits as source of confusion

The main appeal of chaos for cryptographic applications is based on its random-like behavior. Cryptography achieves encryption by embedding the plaintext into a source of entropy. Chaos is a source of entropy. Nevertheless, this source of entropy is conditioned by the dynamics of the specific chaotic system under consideration. Furthermore, there is not only one measure of entropies, but a large set of possible measures. In

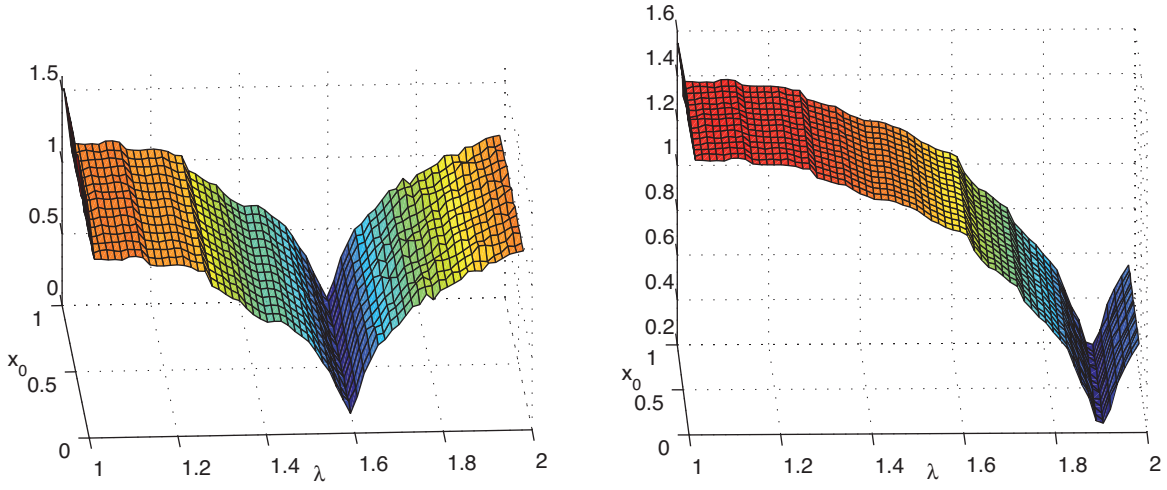


Fig. 2. Wootters' distance of the tent map with respect to the tent map. The length of the binary sequences is $M = 10^4$, whereas the words are of width $w = 10$.

[49, Sec. 2.4] we show a set of measures of entropy, and we analyze unimodal maps by means of those measures. In that work we show that some measures of entropy show a 1-to-1 or 2-to-1 relationship with respect to the control parameter, which could represent a security flaw in the context of chaos-based cryptography.

VI. Analysis of ergodicity

In this section we point out several different critical contexts of chaos-based cryptography where the ergodic behavior of unimodal maps causes security problems.

The first critical context is given by the application of unimodal maps to the design of searching-based chaotic cryptosystems. The efficiency of searching-based chaotic cryptosystems is critically dependent on the invariant probability density function (PDF) of the orbits of the selected chaotic map. The orbits of maps, like the logistic map, the Mandelbrot map, and the tent map, possess a non-uniform PDF, which implies an important increment of the encryption/decryption time. Furthermore, the shape of the PDF of these maps depends on the control parameter(s). In some cryptosystems, as the one described in [36], the diffusion property is compromised by the dependency of the PDF on the control parameter(s). In this sense, we think that the best alternative is the skew tent map, which is a robust chaotic system, i.e., which has a uniform PDF for all the values of the control parameter. Nevertheless, schemes like the one in [36] demand not only a uniform PDF, but also a high LE.

The second critical context is the one drawn by encryption architectures where the ciphertext is obtained by sampling chaotic orbits [26], [50]. In this setting, maps such as the logistic map, the Mandelbrot map, and the tent map should not be used. Indeed, after a transient time all the values derived from the iteration of those maps are inside the interval defined by $[f_\lambda(f_\lambda(x_c)), f_\lambda(x_c)]$ [51], and the histograms of the chaotic orbits show peaks located at different images of the critical point x_c [52]. This means a leak of information about λ that can be used for its estimation, which implies a serious security flaw in the context of chaos-based cryptosystems with ciphertext obtained by sampling chaotic orbits [53], [54]. A way to avoid this critical context is to select chaotic maps with a fixed range for chaotic orbits, which is the case of the skew tent map.

The third critical context is derived from the study of ergodicity by means of order patterns. Suppose that the state space $\mathcal{U}$ is endowed with a total order $<$. Then, the elements of the orbits $\gamma^+_{f_\lambda}(x_0)$ can be arranged from the "smallest" to the "largest" according to the relation $<$. We say that $x \in \mathcal{U}$ defines the order ν-pattern $\pi = [\pi_0, \pi_1, \ldots, \pi_{\nu-1}]$ if $f^{\pi_0}_\lambda(x) < f^{\pi_1}_\lambda(x) < \ldots < f^{\pi_{\nu-1}}_\lambda(x)$. We also say that x is of type π. Observe that $[\pi_0, \pi_1, \ldots, \pi_{\nu-1}]$ is a permutation of the numbers $\{0, 1, ..., \nu - 1\}$. Order patterns can be used to detect determinism [55] and, consequently, to distinguish random systems from chaotic systems. This being the case, the isomorphism between the symbolic dynamics of a chaotic map and a random process does not mean an equivalence by means of order patterns. Actually, there always exist order ν-patterns with sufficiently large ν that are not realized in any orbit of $f \in \mathcal{F}$ [56]. In Fig. 3 the allowed order-4 patterns for the logistic map with $\lambda = 4$ are shown. For this value of the control parameter there exist twelve allowed order patterns, which means a divergence from the twenty-four order patterns of a random system.

Another important application of order patterns is parameter estimation [57]. In general, if f_λ is a family of self-maps of the closed interval $\mathcal{U} \subset \mathbb{R}$ parameterized by $\lambda \in \Lambda \subset \mathbb{R}$ (as it occurs for $f_\lambda \in \mathcal{F}_1, \mathcal{F}_2$), and the set P_π is defined as

$$P_\pi = \{x \in \mathcal{U} : x \text{ is of type } \pi\}, \tag{10}$$

where π is an order $\nu-$pattern, then P_π depends on f_λ and,

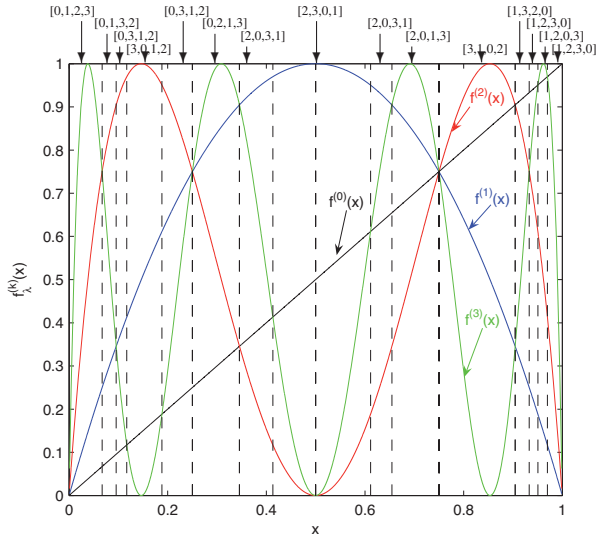


Fig. 3. $f_\lambda^{(k)}(x)$ for $k = 0, 1, 2, 3$ and the corresponding order patterns of length 4 for the logistic map when $\lambda = 4$.

consequently, on λ. Moreover, it is assumed that f_λ is ergodic for $\Lambda \subset \mathbb{R}$ so that the orbits of f_λ can be used to build up statistics independently from the value of the initial condition. According to Birkhoff's ergodic theorem [58, p. 34], if f_λ is ergodic with respect to the invariant measure μ, then the orbit of $x \in \mathcal{U}$ visits the set P_π with relative frequency $\mu(P_\pi)$, for almost all x with respect to μ. As a result, it is possible to study the dependence of P_π on λ by counting and normalizing the occurrences of π in sliding windows of width ν along $\gamma_{f_\lambda}^+(x)$, x being a "typical" initial condition. Let us consider the case of the skew tent map, which possesses a known ergodic invariant measure (the Lebesgue measure) for $\lambda \in (0, 1)$ [59]. As a result, the relative frequency of the order pattern π in a a typical orbit of the skew tent map, coincides with the Lebesgue measure of P_π, which can be determined analytically. For the skew tent map, the interval $P_{[0,1,\ldots,\nu-1]}$ is determined by the leftmost intersection of the iterates $f_\lambda^{\nu-2}$ and $f_\lambda^{\nu-1}$, where

$$f_\lambda^n(x) = \begin{cases} x/\lambda^n, & \text{if } 0 \le x \le \lambda^n, \\ (\lambda^{n-1} - x)/\lambda^{n-1}(1-\lambda), & \text{if } \lambda^n \le x \le \lambda^{n-1}. \end{cases} \tag{11}$$

Hence $P_{[0,1,\ldots,L-1]} = [0, \phi_L(\lambda)]$, with

$$\phi_L(\lambda) = \frac{\lambda^{L-2}}{2-\lambda}. \tag{12}$$

Since this function is 1-to-1 in the interval $0 \le \lambda \le 1$ for $L \ge 2$, with $\phi_2(0) = 1/2$, $\phi_{L\ge 3}(0) = 0$, and $\phi_{L\ge 2}(1) = 1$, it allows to estimate λ by estimating $\phi_L(\lambda)$ —the length of $P_{[0,1,\ldots,L-1]}$ [57].

Order patterns can be used for cryptanalysis when we have access to the whole chaotic orbit or its symbolic edition. This is the case of the scheme described in [35], where encryption is performed through a symbolic sequence of a unimodal map. As we have shown in [44], a chosen-plaintext attack on the cryptosystem defined in [35] can be used to obtain the symbolic sequence used in encryption. If the symbolic sequence was derived from the skew tent map, then the method described in [57] can be used to first determine the order patterns and second to estimate the control parameter.

VII. Conclusion

According to the different analysis shown in this paper, we conclude that unimodal maps possess a large set of vulnerabilities when considering their applications to chaos-based cryptography. However, the identification of different problems of unimodal maps is very constructive with respect to the definition of a framework to design secure and efficient chaos-based cryptosystems. This framework helps figure out how to avoid those critical contexts.

Acknowledgments

The work described in this paper was supported by *Ministerio de Ciencia e Innovación of Spain*, project CUCO (MTM2008-02194) and CODCODS (MTM2009-11820). The work of Shujun Li was supported by a fellowship from the Zukunftskolleg ("Future College") of the University of Konstanz, Germany, which is part of the "Excellence Initiative" Program of the DFG (German Research Foundation).

References

[1] M. Stamp, R. M. Low, Applied cryptanalysis: breaking ciphers in the real world, John Wiley & Sons, Inc., Hoboken, New Jersey, USA, 2007.
[2] R. C. Hilborn, Chaos and nonlinear dynamics, 2nd Edition, Oxford University Press, 2000.
[3] L. Pecora, T. Carroll, Synchronization in chaotic systems, Phys. Rev. Lett. 64 (8) (1990) 821–824.
[4] A. C. Luo, A theory for synchronization of dynamical systems, Communications in Nonlinear Science and Numerical Simulation 14 (5) (2009) 1901 – 1951.
[5] D. Arroyo, C. Li, S. Li, G. Alvarez, W. A. Halang, Cryptanalysis of an image encryption scheme based on a new total shuffling algorithm, Chaos, Solitons and Fractals 41 (5) (2009) 2613–2616.
[6] G. Alvarez, F. Montoya, M. Romera, G. Pastor, Breaking a secure communication scheme based on the phase synchronization of chaotic systems, Chaos 14 (2) (2004) 274–278.
[7] G. Alvarez, F. Montoya, M. Romera, G. Pastor, Cryptanalyzing a discrete-time chaos synchronization secure communication system, Chaos, Solitons & Fractals 21 (3) (2004) 689–694.
[8] G. Alvarez, F. Montoya, M. Romera, G. Pastor, Breaking parameter modulated chaotic secure communication system, Chaos, Solitons & Fractals 21 (4) (2004) 793–797.
[9] G. Alvarez, F. Montoya, M. Romera, G. Pastor, Breaking two secure communication systems based on chaotic masking, IEEE Transactions on Circuits & Systems II 51 (10) (2004) 505–506.
[10] G. Alvarez, S. Li, Breaking network security based on synchronized chaos, Computer Communications 27 (16) (2004) 1679–1681.
[11] G. Alvarez, S. Li, F. Montoya, M. Romera, G. Pastor, Breaking projective chaos synchronization secure communication using filtering and generalized synchronization, Chaos, Solitons & Fractals 24 (3) (2005) 775–783.
[12] S. Li, G. Alvarez, G. Chen, Breaking a chaos-based secure communication scheme designed by an improved modulation method, Chaos, Solitons & Fractals 25 (1) (2005) 109–120.
[13] G. Alvarez, L. Hernández, J. Munoz, F. Montoya, S. Li, Security analysis of communication system based on the synchronization of different order chaotic systems, Physics Letters A 345 (4) (2005) 245–250.
[14] S. Li, G. Alvarez, Z. Li, W. A. Halang, Analog chaos-based secure communications and cryptanalysis: A brief survey, in: The 3rd International IEEE Scientific Conference on Physics and Control (PhysCon 2007), September 3rd-7th 2007 at the University of Potsdam: Abstract Collection, 2007, p. 92, a complete edition available online at http://lib.physcon.ru/?item=1368 and http://www.hooklee.com/Papers/PhysCon2007.pdf.

[15] A. B. Orúe, G. Alvarez, D. Arroyo, J. Nunez, F. Montoya, Determinacin del valor de los parmetros del sistema de Lorenz y aplicacin al criptoanlisis de criptosistemas caticos, in: NoLineal 2007, 2007, p. 85.
[16] A. Orúe, V. Fernandez, G. Alvarez, G. Pastor, M. Romera, F. Montoya, Determination of the parameters for a Lorenz system and application to break the security of two-channel chaotic cryptosystems, Physics Letters A 372 (34) (2008) 5588–5592.
[17] A. Orúe, V. Fernandez, G. Alvarez, G. Pastor, M. Romera, F. Montoya, S. Li, Breaking a SC-CNN-based chaotic masking secure communication system, International Journal of Bifurcation and Chaos 19 (4) (2009) 1329–1338.
[18] M. S. Baptista, Cryptography with chaos, Phys. Lett. A 240 (1-2) (1998) 50–54.
[19] L. Kocarev, G. Jakimoski, Logistic map as a block encryption algorithm, Physics Letters A 289 (2001) 199–206.
[20] G. Jakimoski, L. Kocarev, Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps, IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications 48 (2) (2001) 163–169.
[21] W. Wong, L. Lee, K. Wong, A modified chaotic cryptographic method, Comput. Phys. Comm. 138 (2001) 234–236.
[22] K. W. Wong, A fast chaotic cryptographic scheme with dynamic look-up table, Physics Letters A 298 (2002) 238–242.
[23] N. K. Pareek, V. Patidar, K. Sud, Image encryption using chaotic logistic map, Image and Vision Computing 24 (9) (2006) 926–934.
[24] J. Wei, X. Liao, K. Wong, T. Zhou, Y. Deng, Analysis and improvement for the performance of Baptista's cryptographic scheme, Physics Letters A 354 (2006) 101–109.
[25] J. Wei, X. Liao, K. Wong, T. Xiang, A new chaotic cryptosystem, Chaos, Solitons and Fractals 30 (2006) 1143–1152.
[26] A. N. Pisarchik, N. J. Flores-Carmona, M. Carpio-Valadez, Encryption and decryption of images with chaotic map lattices, Chaos 16 (3) (2006) art. no. 033118.
[27] T. Xiang, X. Liao, G. Tang, Y. Chen, K. wo Wong, A novel block cryptosystem based on iterating a chaotic map, Physics Letters A 349 (2006) 109–115.
[28] T. Gao, Q. Gu, Z. Chen, A new image encryption algorithm based on hyer-chaos, Physics Letters A 372 (4) (2008) 394–400.
[29] Y. Wang, X. Liao, T. Xiang, K.-W. Wong, D. Yang, Cryptanalysis and improvement on a block cryptosystem based on iteration a chaotic map, Physics Letters A 363 (2007) 277–281.
[30] B. W.-K. Ling, C. Y.-F. Ho, P. K.-S. Tam, Chaotic filter bank for computer cryptography, Chaos, Solitons and Fractals 34 (2007) 817–824.
[31] B. Mi, X. Liao, Y. Chen, A novel chaotic encryption scheme based on arithmetic coding 38 (5) (2008) 1523–1531.
[32] H. Yang, X. Lia, K. wo Wong, W. Zhang, P. Wei, A new cryptosystem based on chaotic map and operations algebraic, Chaos, Solitons and Fractals 40 (5) (2009) 2520–2531.
[33] T. Xiang, S. Wang, H. L, G. Hu, A novel symmetrical cryptosystem based on discretized two-dimensional chaotic map, Physics Letters A 364 (2007) 252–258.
[34] X. Wang, Q. Yu, A block encryption algorithm based on dynamic sequences of multiple chaotic systems 14 (2) (2009) 574–581.
[35] A. P. Kurian, S. Puthusserypady, Self-synchronizing chaotic stream ciphers, Signal Processing 88 (2008) 2442–2452.
[36] A. N. Pisarchik, M. Zanin, Image encryption with chaotically coupled chaotic maps, Physica D 237 (2008) 2638–2648.
[37] H.-O. Peitgen, H. Jurgens, D. Saupe, Chaos and Fractals, SpringerVerlag, 1992.
[38] C. Abraham, G. Biau, B. Cadre, On Lyapunov exponent and sensitivity, Journal of Mathematical Analysis and Applications 290 (2004) 395–404.
[39] W. Tucker, D. Wilczak, A rigorous lower bound for the stability regions of the quadratic map, Physica D: Nonlinear Phenomena 238 (18) (2009) 1923–1936.
[40] S. Banerjee, J. A. Yorke, C. Grebogi, Robust chaos, Physical Review Letters 80 (1998) 3049–3052.
[41] J. M. Aguirregabiria, Robust chaos with variable Lyapunov exponent in smooth one-dimensional maps, Chaos, Solitons and Fractals 42 (2009) 2531–2539.
[42] J. M. Amigó, L. Kocarev, J. Szczepanski, On some properties of the discrete Lyapunov exponent, Physics Letters A (2008) 6265–6268.
[43] A. P. Majtey, P. W. Lamberti, M. T. Martin, A. Plastino, Wootters' distance revisited: a new distinguishability criterium, Eur. Phys. J. D 32 (2005) 413–419.
[44] D. Arroyo, G. Alvarez, J. M. Amigó, S. Li, Cryptanalysis of a family of self-synchronizing chaotic stream ciphers, Communications in Nonlinear Science and Numerical Simulation, Accepted April 23.
[45] N. Metropolis, M. Stein, P. Stein, On the limit sets for transformations on the unit interval, Journal of Combinatorial Theory (A) 15 (1973) 25–44.
[46] G. Alvarez, F. Montoya, M. Romera, G. Pastor, Cryptanalysis of an ergodic chaotic cipher, Physics Letters A 311 (2003) 172–179.
[47] X. Wu, H. Hu, B. Zhang, Parameter estimation only from the symbolic sequences generated by chaos system, Chaos, solitons and Fractals 22 (2004) 359–366.
[48] D. Arroyo, G. Alvarez, S. Li, C. Li, V. Fernandez, Cryptanalysis of a new chaotic cryptosystem based on ergodicity, International Journal of Modern Physics B 23 (5) (2009) 651–659.
[49] D. Arroyo, Framework for the analysis and design of encryption strategies based on discrete-time chaotic dynamical systems, Ph.D. thesis, ETSIA of the Polytechnic University of Madrid, Madrid, Spain, avalaible online at http://digital.csic.es/handle/10261/15668 (July 2009).
[50] E. Alvarez, A. Fernández, P. García, J. Jiménez, A. Marcano, New approach to chaotic encryption, Physics Letters A 263 (1999) 373–375.
[51] J. Guckenheimer, Sensitive dependence to initial conditions for one dimensional maps, Communications in Mathematical Physics 70 (2) (1979) 133–160.
[52] R. V. Jensen, C. R. Myers, Images of the critical point of nonlinear maps, Physical Review A 32 (2) (1985) 1222–1224.
[53] G. Alvarez, F. Montoya, M. Romera, G. Pastor, Cryptanalysis of a chaotic encryption system, Physics Letters A 276 (2000) 191–196.
[54] D. Arroyo, R. Rhouma, G. Alvarez, S. Li, V. Fernandez, On the security of a new image encryption scheme based on chaotic map lattices, Chaos: An Interdisciplinary Journal of Nonlinear Science 18 (2008) 033112, 7 pages.
[55] J. M. Amigó, L. Kocarev, J. Szczepanski, Order patterns and chaos, Physics Letters, Section A: General, Atomic and Solid State Physics 355 (1) (2006) 27–31.
[56] J. M. Amigó, S. Elizalde, M. B. Kennel, Forbidden patterns and shift systems, Journal of Combinatorial Theory, Series A 115 (2008) 485–504.
[57] D. Arroyo, G. Alvarez, J. M. Amigó, Estimation of the control parameter from symbolic sequences: Unimodal maps with variable critical point, Chaos: An Interdisciplinary Journal of Nonlinear Science 19 (2009) 023125, 9 pages.
[58] P. Walters, An Introduction to Ergodic Theory, Vol. 79 of Graduate Texts in Mathematics, Springer-Verlag, New York, 1982.
[59] L. Billings, E. M. Bollt, Probability density functions of some skew tent maps, Chaos, solitons and fractals 12 (2) (2001) 365–376.

Cifrado de flujo con autómatas celulares difusos

Francisco José Navarro-Ríos
Computer and Communications Service Center, University of Granada, 18071 Granada, Spain
Email: francisco@ugr.es

***Resumen*—Este trabajo propone, por primera vez en la literatura, la utilización del modelo de cálculo teórico conocido como "autómatas celulares difusos" para realizar cifrados de información. Para ello, creamos nuevos operadores difusos, diseñamos nuevas funciones locales de transición para los autómatas, desarrollamos un software que haga evolucionar los autómatas, que realice cifrados de flujo, que cifre y descifre utilizando una nueva función real con carácter involutivo, y que compruebe la bondad de este nuevo sistema de cifrado mediante diversos tests de pseudoaleatoriedad sobre las secuencias cifrantes generadas con las evoluciones de los autómatas, y sobre los criptogramas obtenidos, y mediante análisis de la correlación entre textos en claro y sus criptogramas.**

I. Introducción

El campo del conocimiento que trata la *Criptología* es prácticamente tan antiguo como la escritura, y ha evolucionado junto con el resto de los desarrollos teóricos y de ingeniería que el hombre ha ido realizando ([1], [2]). La búsqueda de sistemas de cifrado criptográficamente seguros es uno de los problemas más interesantes y abiertos que tienen planteados conjuntamente las Ciencias de la Computación, el Álgebra y la Matemática Aplicada. El enfoque orientado al modelo matemático de los autómatas celulares convencionales (no difusos) es relativamente novedoso, y proporciona un espacio de desarrollo amplio y de múltiples posibilidades.

Los *autómatas celulares* son introducidos por John von Neumann en los años 60 [3], y reciben un gran impulso en los años 80, cuando Stephen Wolfram los utiliza sistemáticamente como modelos representativos de sistemas dinámicos ([4], [5], [6]). La literatura recoge un autómata celular concreto, presentado por Stephen Wolfram en 1986 [5], con el cual se implementa un sistema de cifrado muy simple. Durante los años 90 y, hasta ahora, se publican multitud de trabajos que implementan cifrados basados en autómatas celulares: Howard Gutowitz (1993) [7], Nandi, Kar y Chaudhuri (1994) [8], Lafe (1996) [9], Sen, Shaw, Chowdhuri, Ganguly y Chaudhuri (2002) [10], Amparo Fúster y Dolores de la Guía (2007) [11], etc.

Por otra parte, la teoría de *conjuntos difusos* o borrosos se introduce en 1965 [12], con el célebre trabajo llamado "Fuzzy Sets", de Lotfi Zadeh. Los *números difusos* empiezan a tener importancia con el trabajo de D. Dubois y H. Prade [13], sobre variables difusas y el manejo de cantidades imprecisas. Los conjuntos difusos fueron aplicados por primera vez en la teoría de autómatas por W.G. Wee y K.S. Fu, a finales de los años 60 [14], dando lugar a los *autómatas celulares difusos*.

Los autómatas celulares difusos ya han sido utilizados con éxito en aplicaciones para la simulación de sistemas naturales, como, por ejemplo, la expansión del fuego en un bosque, y también en aplicaciones para la simulación de sistemas artificiales, como, por ejemplo, el desarrollo del tráfico en una ciudad, y en otras aplicaciones similares, por autores como Mingarelli en 2007 [15], Zhang, Li y Zhao en 2007 [16], Maji y Chaudhuri en 2007 [17], Mandelas, Hatzichristos y Prastacos en 2007 [18], Maji en 2008 [19], Basu y Basu en 2008 [20], Yacoubi y Mingarelli en 2008 [21], etc. Sin embargo, la aplicación del modelo difuso al ámbito del desarrollo de protocolos criptográficos es una cuestión aún no tratada en la comunidad científica.

II. Definición de nuevos operadores difusos

Desde que se definieron por primera vez los autómatas celulares difusos por W.G. Wee y K.S. Fu, a finales de los años 60 [14], y hasta nuestros días, con investigadores como Mingarelli, Zhang, Maji, etc. ([15]–[21]), se han utilizado habitualmente tres operadores difusos en los autómatas celulares: la suma difusa ( ), el producto difuso ( ) y el complemento difuso ( ), según se definen en el Cuadro I mediante los operadores aritméticos convencionales de suma ( ), resta ( ) y producto ( ), donde y son números difusos: números reales del intervalo cerrado .

Cuadro I
Definición de operadores difusos con operadores aritméticos

Operadores difusos	Operadores aritméticos

Sin embargo, para nuestros propósitos criptográficos, vamos a definir dos nuevos operadores difusos, sobre los números reales del intervalo abierto : la suma difusa módulo uno ( ), y el complemento difuso módulo uno ( ), mediante las Funciones (1) y (2), respectivamente:

$$\begin{array}{l} \text{si} \\ \text{si} \end{array} \quad (1)$$

$$\begin{array}{l} \text{si} \\ \text{si} \end{array} \quad (2)$$

Utilizaremos estos nuevos operadores difusos tanto en el diseño de funciones locales de transición para la evolución de autómatas, como en el diseño de una función real con carácter involutivo para el cifrado y descifrado de información, que tenga un comportamiento similar a la XOR clásica de los criptosistemas de flujo.

III. Diseño de una nueva función real involutiva

En 1917, J. Mauborgne y G. Vernam inventaron el cifrado de flujo. Desde entonces, se ha utilizado la función XOR (la suma binaria módulo dos) como algoritmo de cifrado y descifrado de información. En los años 90 [8], también se ha estado usando la función XNOR (su complementaria). Estos operadores trabajan sobre números binarios, o sea, se aplican bit a bit sobre la secuencia del mensaje (texto en claro), la clave (secuencia cifrante) y el criptograma (texto cifrado), y tienen la característica de ser involutivas.

Los operadores XOR y XNOR funcionan muy bien con autómatas celulares booleanos [8], pero, en este artículo, pretendemos trabajar con autómatas celulares difusos, así que uno de nuestros objetivos es encontrar o diseñar una función que se aplique en el intervalo real y que tenga este comportamiento involutivo:

(3)

donde . Finalmente, diseñamos la siguiente función de cifrado y de descifrado, y comprobamos su buen funcionamiento:

(4)

donde es cada número difuso del texto en claro, es cada número difuso de la secuencia cifrante, es cada número difuso del criptograma, es la suma difusa módulo uno, y es el complemento difuso módulo uno, definidos en las Funciones (1) y (2), respectivamente.

IV. Diseño de una función de conversión difusa

Los mensajes, las secuencias cifrantes y los criptogramas van a ser cadenas de números difusos. Para convertir un texto en claro (formado por caracteres) en una secuencia de números difusos, calculamos el grado de pertenencia de cada carácter al conjunto difuso de los 256 caracteres ASCII mediante la siguiente función que hemos diseñado:

——— (5)

donde es la función que devuelve el valor decimal del código ASCII del carácter . Los valores que tiene son los números naturales que van desde 0 hasta 255.

De esta forma, adquiere valores de números reales equidistantes en el intervalo abierto . Decidimos evitar que sea 1, porque ese valor no está definido para la función de cifrado descrita en (4). Tampoco vamos a utilizar el 0, ya que la función de cifrado devuelve el criptograma 0 cuando se usa la clave 0 para cifrar el número difuso 0 (*clave débil*).

Por otra parte, cuando hayamos realizado el descifrado del criptograma, y queramos obtener el texto en claro original a partir del texto difuso descifrado, aplicaremos la siguiente fórmula para calcular el valor decimal del código ASCII de cada carácter del texto en claro original:

(6)

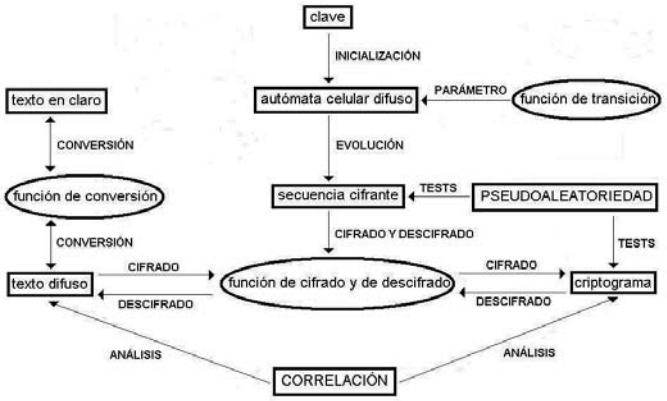


Figura 1. Diagrama descriptivo de los procesos de cifrado y de análisis

V. Diseño de autómatas celulares difusos

Vamos a trabajar con autómatas celulares unidimensionales, y con condición de frontera cilíndrica. Para hacer las pruebas iniciales de pseudoaleatoriedad y de correlación, en principio hemos parametrizado una longitud del autómata de 128 celdas, un tamaño del mensaje de 1000 caracteres (ó 1000 evoluciones del autómata), y un radio igual a uno para el vecindario, para varias (trece) funciones locales de transición diferentes que hemos diseñado y que exponemos en el Cuadro II, donde , y son la celda anterior, la celda actual y la celda siguiente, respectivamente.

Cuadro II
Diseño de funciones locales de transición

Función	Diseño con operadores difusos
1	
2	0
3	0
4	
5	0
6	0
7	
8	0
9	0
10	
11	0
12	0
13	0

Los cuatro operadores que se utilizan en estas funciones locales de transición son los cuatro operadores difusos descritos en el Cuadro I y en la Función (1). El orden de preferencia de los operadores es: primero el complemento ( ), después el producto ( ) y, por último, las sumas (,). Estas dos sumas tienen la misma preferencia.

Observaremos, por ejemplo, la primera de las dos celdas centrales, para la creación de la secuencia cifrante (hay dos celdas centrales, puesto que el autómata tiene longitud par). Si no hay presencia de pseudoaleatoriedad en la evolución de una de las celdas, difícilmente habrá pseudoaleatoriedad en las demás celdas del autómata, debido a las dependencias.

VI. Análisis de aleatoriedad y de correlación

Para comprobar la bondad del nuevo criptosistema, vamos a pasar varios tests de aleatoriedad a las secuencias cifrantes y a los criptogramas obtenidos:

- Test 1. Chi-Cuadrado para muestras del 50 %:
  [0, 0.5[y [0.5, 1].
- Test 2. Chi-Cuadrado para muestras del 25 %:
  [0, 0.25[, [0.25, 0.5[, [0.5, 0.75[y [0.75, 1].
- Test 3. Chi-Cuadrado para muestras del 10 %:
  [0.0, 0.1[, [0.1, 0.2[, [0.2, 0.3[, [0.3, 0.4[, [0.4, 0.5], [0.5, 0.6[, [0.6, 0.7[, [0.7, 0.8[, [0.8, 0.9[, [0.9, 1.0].
- Test 4. Frecuencias de los números difusos y .
- Test 5. Frecuencias de series de dos números difusos: , , y .
- Test 6. Frecuencias de series de tres números difusos: , , , , , y .

En los tres últimos tests, denotamos a los números difusos del intervalo [0, 0.5[, y a los números difusos del intervalo [0.5, 1] de la secuencia cifrante y de la cifrada.

No vamos a mostrar los resultados del Test 4, porque coinciden con los del Test 1, puesto que se cumple la igualdad donde se define el cálculo de cada test de aleatoriedad:

$$\frac{\qquad\qquad}{\qquad} \qquad (7)$$

donde es la longitud de la secuencia cifrante, es la cantidad esperada, y donde se cumplen las siguientes ecuaciones:

$$- \qquad (8)$$

Por otra parte, para calcular la posible dependencia entre textos en claro y sus criptogramas, usaremos el coeficiente de correlación de Spearman ( ).

VII. Resultados y discusión

Para probar cada una de las trece funciones locales de transición diseñadas, el software (desarrollado en lenguaje C++) va mostrando en la pantalla del ordenador todo el proceso paso a paso: inicialización pseudoaleatoria del autómata celular difuso (con números reales generados por el programa), evolución del autómata mil veces, creación de la secuencia cifrante, y tests de aleatoriedad sobre la secuencia cifrante, con los resultados que se muestran en el Cuadro III, redondeados a dos decimales.

Interesa que los resultados de todos los tests de aleatoriedad estén próximos a cero para que haya pseudoaleatoriedad. Para una secuencia cifrante de 1000 números difusos, el valor máximo de cada test es 1000 para el Test 1, 3000 para el Test 2, 9000 para el Test 3, 1998 para el Test 5, y 2331 para el Test 6.

En el Cuadro III, se observa que las seis primeras funciones de transición no superan ningún test de aleatoriedad, porque tienen estadísticos muy elevados (igual o por encima de mil). Esto tiene una explicación lógica: si observamos la evolución del autómata en el programa, veremos que todas las celdas

Cuadro III
Tests de aleatoriedad sobre la secuencia cifrante para las trece funciones de transición diseñadas

Función	Test 1	Test 2	Test 3	Test 5	Test 6
1	1000	2992	8980	1997	2331
2	1000	2992	8980	1997	2331
3	1000	3000	8980	1997	2331
4	1000	3000	9000	1997	2331
5	1000	2992	8980	1997	2331
6	1000	3000	9000	1997	2331
7	0	1000	4000	999	999
8	1.6	842	3213	847	810
9	0.68	870	3483	898	879
10	0.58	2.51	4.24	0.60	3.75
11	3.36	7.98	13.26	3.49	7.40
12	0.68	1.27	10.74	0.58	4.03
13	1.44	2.47	5.44	2.67	2.02

tienden al estado difuso cero en el caso de las tres primeras funciones (que operan con el producto difuso) y al estado difuso uno en el caso de las tres funciones siguientes (que operan con la suma difusa).

Las siguientes tres funciones de transición (7, 8, y 9) trabajan con el complemento difuso, y podemos observar, en el programa, que producen una secuencia cifrante que alterna entre ceros y unos, por eso superan el primer test, pero ninguno de los demás tests, así que estas funciones no nos sirven.

Las únicas funciones de transición que superan todos los tests de aleatoriedad son precisamente las cuatro últimas del Cuadro II, que trabajan con el nuevo operador difuso que hemos diseñado en este artículo. Por consiguiente, acabamos de lograr otro de los objetivos del trabajo: diseñar alguna función de transición que genere una secuencia cifrante que supere los tests de aleatoriedad. De las cuatro, vamos a elegir, por ejemplo, para seguir con la investigación, la Función 10, que tiene todos los estadísticos por debajo de cinco. Para asegurarnos más de la bondad de esta función de transición, ejecutamos el software varias (dieciséis) veces, utilizando la Función 10 para hacer evolucionar autómatas, que se inicializan aleatoriamente en cada ejecución del programa, y obtenemos buenos resultados: todos por debajo de cinco para el Test 1, por debajo de ocho para el Test 2, por debajo de veintitrés para el Test 3, por debajo de siete para el Test 5, y por debajo de quince para el Test 6, según el Cuadro IV.

El siguiente paso consiste en utilizar las secuencias cifrantes que genera el autómata celular difuso con la nueva función de transición que hemos diseñado (la Función 10) para cifrar una batería de textos en claro, a través de la función real involutiva que diseñamos, con la intención de medir la correlación entre los mensajes y los criptogramas producidos, y también comprobar la aleatoriedad de dichos criptogramas, aplicándoles los mismos tests de aleatoriedad que usamos anteriormente para las secuencias cifrantes. Para medir la correlación, hemos utilizado el coeficiente de correlación de Spearman ( ), y hemos seleccionado siete textos en claro diferentes en español (filtrados y truncados a mil caracteres). Se trata de los primeros mil caracteres de los primeros siete capítulos de la tesis doctoral del autor de este artículo. El software que hemos

Cuadro IV
TESTS DE ALEATORIEDAD SOBRE LA SECUENCIA CIFRANTE PARA LA FUNCIÓN DE TRANSICIÓN 10 SELECCIONADA

Ejecución	Test 1	Test 2	Test 3	Test 5	Test 6
1	1.44	2.22	18.36	6.94	14.12
2	0.78	1.08	7.14	3.60	5.43
3	0.48	1.27	9.68	1.02	6.58
4	0.02	4.4	5.04	0.10	3.17
5	0.48	2.42	6.72	0.49	9.37
6	4.62	5.79	22.86	6.16	9.85
7	0.02	3.34	9.42	0.10	7.35
8	0.68	1.14	13.26	0.69	5.38
9	0.26	0.74	7.74	0.26	7.59
10	1.94	1.98	16.06	2.67	7.64
11	0.1	2.89	15.38	0.37	4.42
12	0.26	0.54	1.46	0.64	4.47
13	1.16	1.88	8.14	3.05	2.59
14	4.62	7.73	16.24	4.37	8.45
15	0.00	1.94	9.12	0.09	2.26
16	0.06	2.78	1.28	3.06	4.42

desarrollado se encarga de ir contando los caracteres de un texto mayor hasta alcanzar los mil caracteres que tenemos como parámetro, y también de ir realizando un filtrado para eliminar caracteres especiales que vayan a causar problemas (debido a que no se corresponden con ningún valor ASCII entre 0 y 255). Los resultados se muestran en el Cuadro V.

Cuadro V
CORRELACIÓN, Y TESTS DE ALEATORIEDAD SOBRE EL CRIPTOGRAMA, PARA SIETE TEXTOS DISTINTOS, CON LA FUNCIÓN DE TRANSICIÓN 10

Mensaje	Correlación	Test 1	Test 2	Test 3	Test 5	Test 6
Texto 1	–0.020697	7.40	8.70	10.74	8.60	19.22
Texto 2	–0.020510	1.44	3.48	14.28	1.81	13.02
Texto 3	–0.036655	0.9	4.94	10.92	2.88	5.81
Texto 4	–0.036660	0.9	2.98	5.42	0.91	6.96
Texto 5	–0.030538	0.4	3.99	8.5	6.73	15.18
Texto 6	–0.000387	0.9	4.97	12.44	1.39	7.30
Texto 7	0.038301	0.58	4.17	13.18	1.31	6.00

Los resultados del Cuadro V son buenos, porque, para todos los textos, el coeficiente de correlación es próximo a cero, y todos los tests de aleatoriedad están por debajo de 20, así que hemos obtenido otro de los objetivos del trabajo: que no hubiera correlación entre mensajes y criptogramas con nuestro nuevo cifrado, y que los criptogramas superen los tests.

Finalmente, el software muestra cómo desciframos cada uno de los siete criptogramas con la función real involutiva que diseñamos y con la misma secuencia cifrante, obteniendo de nuevo los siete textos en claro originales.

A modo de ejemplo de lo que serían unos malos estadísticos en este caso, comprobamos los resultados de cifrar el Texto 1 con todas las funciones de transición que desechamos, pero hay que tener en cuenta que realmente jamás se debe cifrar con este tipo de funciones (cuyas secuencias cifrantes no superaron los tests de aleatoriedad), sino que esto es sólo para observar qué pasaría si lo hiciésemos, y que veamos la gran diferencia que hay en los estadísticos: coeficientes de correlación próximos a –1 (máximo valor de dependencia negativa), y todos los tests de aleatoriedad por encima de 800, según se puede observar en el Cuadro VI.

Cuadro VI
CORRELACIÓN, Y TESTS DE ALEATORIEDAD SOBRE EL CRIPTOGRAMA, PARA EL TEXTO 1, CON LAS NUEVE PRIMERAS FUNCIONES

Función	Correlación	Test 1	Test 2	Test 3	Test 5	Test 6
1	–0.980174	996	1782	2747	1993	2315
2	–0.977292	996	1782	2747	1993	2315
3	–0.980032	1000	1789	2758	1997	2331
4	–0.984420	1000	1789	2743	1997	2331
5	–0.984200	1000	1784	2743	1997	2331
6	–0.984420	1000	1784	2743	1997	2331
7	–0.986696	1000	1789	2743	1997	2331
8	–0.825714	810	1473	2275	1560	1725
9	–0.848751	872	1604	2426	1687	1906

Un aspecto fundamental en los cifrados simétricos es la *longitud de la clave privada*. Hasta ahora, hemos estado utilizando 128 números difusos para inicializar los autómatas, pero debemos investigar si se pueden obtener secuencias cifrantes pseudoaleatorias a partir de claves de menor tamaño.

Cuadro VII
MEDIAS DE LOS TESTS DE ALEATORIEDAD PARA 2000 EJECUCIONES

Clave	Secuencia	Test 1	Test 2	Test 3	Test 5	Test 6
128	28000	1.00	3.00	9.01	2.01	6.97
128	18000	1.06	3.10	9.13	2.01	7.19
128	8000	0.99	2.99	8.93	1.94	6.94
128	1000	1.06	3.04	9.15	2.03	7.13
64	28000	0.98	2.91	8.94	1.96	6.95
64	18000	1.02	3.14	9.37	2.03	6.97
64	8000	0.98	3.10	9.12	2.00	7.00
64	1000	0.97	2.97	8.83	1.98	6.99
32	28000	0.93	2.90	8.88	1.89	6.93
32	18000	1.03	3.05	9.18	2.05	7.12
32	8000	0.99	2.96	8.90	2.00	6.97
32	1000	1.02	3.10	9.14	2.01	7.03
16	28000	0.98	2.97	8.97	1.97	6.96
16	18000	1.04	3.08	9.06	2.00	6.96
16	8000	1.04	3.12	9.07	2.08	7.10
16	1000	1.00	3.01	9.08	1.99	6.91
8	28000	0.99	3.00	8.88	2.02	7.14
8	18000	1.02	2.98	8.99	2.03	7.14
8	8000	1.06	3.15	9.18	2.06	6.94
8	1000	1.00	3.03	9.05	2.01	7.12
7	28000	0.98	2.94	9.03	2.00	6.93
7	18000	1.01	3.03	9.04	1.99	6.95
7	8000	0.98	3.04	8.94	2.00	6.97
7	1000	1.00	2.94	8.86	1.98	7.01
6	28000	1.06	3.09	9.07	1052.60	746.74
6	18000	1.01	3.03	9.06	664.25	473.66
6	8000	0.97	2.97	8.95	294.61	213.56
6	1000	0.97	2.95	8.85	34.75	30.07
5	28000	0.98	2.94	8.99	2.02	7.00
5	18000	1.01	2.97	8.85	2.00	7.06
5	8000	1.00	3.05	9.24	1.93	6.99
5	1000	0.99	3.01	8.99	2.00	7.07
4	28000	1.01	3.08	9.22	520.24	371.89
4	18000	1.04	3.06	9.21	334.53	240.79
4	8000	1.01	3.03	9.02	150.43	111.03
4	1000	0.98	2.99	8.99	20.63	19.87
3	28000	2.06	4.68	10.28	3113.34	2195.32
3	18000	2.06	4.55	10.23	2001.53	1415.05
3	8000	2.01	4.51	10.17	891.23	633.65
3	1000	2.00	4.50	10.24	114.28	85.53
2	28000	1.00	3.11	9.15	1037.07	735.18
1	28000	16.08	46.59	136.43	3138.50	2225.31

Cuadro VIII
MÁXIMOS DE LOS TESTS DE ALEATORIEDAD PARA 2000 EJECUCIONES

Clave	Secuencia	Test 1	Test 2	Test 3	Test 5	Test 6
128	28000	16.22	19.69	27.48	16.78	25.32
128	18000	16.81	26.93	30.61	19.33	29.84
128	8000	11.40	21.32	35.27	17.91	28.92
128	1000	11.24	16.34	30.08	19.47	27.34
64	28000	11.85	21.27	31.78	13.78	25.53
64	18000	15.02	17.76	31.16	16.92	34.64
64	8000	11.70	22.49	31.96	14.66	24.92
64	1000	11.66	18.14	28.48	12.70	26.38
32	28000	13.38	17.15	30.62	13.40	25.55
32	18000	12.59	16.85	29.24	15.30	27.04
32	8000	11.55	20.09	29.79	14.12	30.17
32	1000	12.1	20.78	34.68	14.82	25.41
16	28000	14.00	18.07	30.85	15.27	29.16
16	18000	13.01	17.80	40.13	13.57	24.61
16	8000	14.96	24.88	31.76	17.47	23.30
16	1000	12.1	21.32	29.46	16.73	26.18
8	28000	11.93	17.94	28.99	15.70	33.35
8	18000	13.78	15.58	26.73	17.51	28.01
8	8000	15.66	23.15	32.82	16.13	27.71
8	1000	10	14.90	30.72	14.39	29.45
7	28000	13.20	14.18	26.82	13.75	23.50
7	18000	13.23	21.53	31.54	13.79	23.76
7	8000	14.96	23.81	28.60	15.39	23.87
7	1000	10.82	16.49	33.08	15.12	26.66
6	28000	15.00	20.15	34.64	3339.05	2312.48
6	18000	18.95	23.00	34.75	2201.62	1534.32
6	8000	20.40	24.89	42.78	1017.66	728.36
6	1000	19.6	19.98	36.26	144.27	121.08
5	28000	11.04	15.21	27.91	14.32	31.06
5	18000	14.00	15.78	31.19	15.27	24.37
5	8000	15.66	19.04	28.89	15.75	30.52
5	1000	9.60	19.21	33.72	16.40	28.30
4	28000	15.84	21.29	39.62	2918.62	2061.69
4	18000	12.48	19.91	31.37	1970.24	1401.25
4	8000	14.45	23.51	35.41	860.50	643.23
4	1000	14.4	20.79	34.4	137.12	124.68
3	28000	26.29	33.73	39.76	3560.70	2577.08
3	18000	24.64	28.71	42.32	2247.75	1665.27
3	8000	21.42	27.39	47.12	1066.71	813.25
3	1000	24.34	31.78	42.98	189.62	167.92
2	28000	12.43	24.96	33.53	3300.77	2470.18
1	28000	28000	84000	252000	55997	65331

Cuadro IX
MÍNIMOS DE LOS TESTS DE ALEATORIEDAD PARA 2000 EJECUCIONES

Clave	Secuencia	Test 1	Test 2	Test 3	Test 5	Test 6
128	28000	0	0.01	1.07	0.00	0.62
128	18000	0	0.01	0.79	0.00	0.53
128	8000	0	0.01	0.62	0.00	0.42
128	1000	0	0.01	1.02	0.00	0.14
64	28000	0	0.01	0.40	0.00	0.26
64	18000	0	0.00	0.80	0.00	0.78
64	8000	0	0.01	0.96	0.00	0.61
64	1000	0	0	1.14	0.00	0.53
32	28000	0	0.01	0.52	0.00	0.39
32	18000	0	0.01	0.67	0.00	0.20
32	8000	0	0.01	0.88	0.01	0.53
32	1000	0	0.02	0.96	0.00	0.57
16	28000	0	0.02	0.82	0.00	0.62
16	18000	0	0.01	0.80	0.00	0.59
16	8000	0	0.01	1.31	0.00	0.75
16	1000	0	0.06	1.24	0.00	0.43
8	28000	0	0.01	0.89	0.00	0.54
8	18000	0	0.00	1.10	0.00	0.25
8	8000	0	0.02	0.78	0.00	0.35
8	1000	0	0.01	0.98	0.00	0.72
7	28000	0	0.00	0.39	0.00	0.63
7	18000	0	0.03	0.90	0.00	0.56
7	8000	0	0.01	1.03	0.00	0.17
7	1000	0	0.01	0.54	0.00	0.43
6	28000	0	0.02	0.37	0.01	56.44
6	18000	0	0.03	0.87	0.03	31.78
6	8000	0	0.01	0.85	0.04	11.73
6	1000	0	0.02	1.04	0.00	0.96
5	28000	0	0.01	0.95	0.00	0.60
5	18000	0	0.00	0.79	0.00	0.49
5	8000	0	0.01	0.89	0.00	0.22
5	1000	0	0.01	1.2	0.00	0.48
4	28000	0	0.02	1.20	0.02	3.65
4	18000	0	0.03	1.17	0.01	1.22
4	8000	0	0.00	1.01	0.00	1.62
4	1000	0	0.02	0.7	0.01	0.72
3	28000	0	0.01	0.81	2686.60	1844.76
3	18000	0	0.01	0.60	1718.02	1180.92
3	8000	0	0.01	0.49	713.60	472.39
3	1000	0	0.01	0.58	48.59	32.05
2	28000	0	0.02	0.87	0.05	51.39
1	28000	0	0.04	0.66	2803.05	1920.97

Para determinar qué longitud de clave sería la mínima necesaria, parametrizamos el programa con distintos tamaños de autómata (128, 64, 32, 16, 8, 7, 6, 5, 4, 3, 2, 1) y con distintos tamaños de secuencias cifrantes (1000, 8000, 18000, 28000), y ejecutamos 2000 veces cada combinación de longitud de autómata y longitud de secuencia cifrante, con diferentes inicializaciones pseudoaleatorias de los autómatas, para calcular la media, el máximo y el mínimo de cada grupo de 2000 ejecuciones, con los resultados que se muestran en los Cuadros VII, VIII y IX.

Se obtienen unos resultados excelentes y similares para claves de 5 caracteres (o 5 números difusos), y para claves de 7 caracteres o más (8, 16, 32, 64, 128). Las claves de 6, 4, 3 y 2 caracteres superan los tres tests Chi-Cuadrado, pero no los dos tests de series, así que no nos sirven. Y la clave de un sólo carácter no supera ningún test de aleatoriedad. No obstante, deberemos continuar la investigación implementando tests de aleatoriedad más sofisticados, y utilizando muestras mayores.

VIII. CONCLUSIONES

Hemos cumplido el objetivo fundamental de este trabajo, que consistía en realizar cifrados de flujo utilizando el modelo de cálculo teórico conocido como "autómatas celulares difusos", y en comprobar la bondad del sistema de cifrado mediante diversos tests de aleatoriedad y análisis de correlación.

Para desarrollar este criptosistema, hemos tenido que definir nuevos operadores de números difusos, diseñar una función real en el dominio de los números difusos que tenga carácter involutivo para realizar el cifrado y el descifrado (que utiliza los nuevos operadores difusos), diseñar una función local de transición para autómatas celulares difusos que genere secuencias cifrantes que superen los tests de aleatoriedad (que también utiliza uno de los nuevos operadores difusos), y desarrollar un software (en lenguaje C++) que realiza las siguientes tareas: inicialización del autómata celular difuso (pseudoaleatoria o con clave introducida por el usuario), evolución del autómata con funciones de transición, creación

de la secuencia cifrante, tests de aleatoriedad sobre la secuencia cifrante, carga del mensaje (texto en claro), filtro de eliminación de caracteres especiales, conversión del mensaje a números difusos, comprobación del carácter involutivo de funciones reales, cifrado del mensaje con la secuencia cifrante, cálculo de correlación entre mensaje y criptograma, tests de aleatoriedad sobre el criptograma, descifrado del criptograma con la secuencia cifrante, y chequeo de la integridad entre el descifrado y el texto original.

Así que hemos comprobado la utilidad de los autómatas celulares difusos en el cifrado de flujo, y la viabilidad técnica para su integración e implementación en protocolos criptográficos.

No obstante, deberemos continuar la investigación para determinar qué ofrece este nuevo cifrado de flujo, frente al habitual en términos binarios, en cuanto a rapidez, seguridad, facilidad de implementación software y hardware, longitud de la clave, mejores prestaciones, posibles problemas o inconvenientes a la hora de su incorporación masiva al mundo criptográfico, etc.

REFERENCIAS

[1] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, "Handbook of Applied Cryptography", *CRC Press*, 1996.

[2] B. Schneier, "Applied Cryptography", Second Edition, *John Wiley & Sons*, 1996.

[3] J. von Neumann, "Theory of Self-reproducing Automata", *Press Urbana*, University of Illinois, 1966.

[4] S. Wolfram, "Cellular Automata", *Los Alamos Science*, 9, pp. 2–21, 1983.

[5] S. Wolfram, "Theory and Applications of Cellular Automata", *World Scientific*, 1986.

[6] S. Wolfram, "Cellular Automata and Complexity: Collected Papers", *Perseus Publishing*, 1993.

[7] H. Gutowitz, "A Massively Parallel Cryptosystem Based on Cellular Automata", *ESPCI*, Paris, France, 1993.

[8] S. Nandi, B.K. Kar, P.P. Chaudhuri. "Theory and Applications of Cellular Automata in Cryptography", *Proceedings of the IEEE Transactions on Computers*, vol. 43, no. 12, pp. 1346–1357, December 1994.

[9] O. Lafe, "Data Compression and Encryption using Cellular Automata Transforms", *Proceedings of the IEEE International Joint Symposia on Intelligence and Systems (IJSIS)*, pp. 234–241, 1996.

[10] S. Sen, C. Shaw, D.R. Chowdhuri, N. Ganguly, P.P. Chaudhuri, "Cellular Automata based Cryptosystem", *ICICS*, pp. 303–314, Singapore, December 2002.

[11] A. Fúster, D. de la Guía, "Modelling Nonlinear Sequence Generators in Terms of Linear Cellular Automata", *Applied Mathematical Modelling*, vol. 31, no. 2, pp. 226–235, 2007.

[12] L.A. Zadeh, "Fuzzy Sets", *Information and Control*, 83, pp. 338–353, 1965.

[13] D. Dubois, H. Prade, "Theorie des Possibilities", *Masson*, Paris, France, 1985.

[14] W.G. Wee, K.S. Fu, "A Formulation of Fuzzy Automata and its Application as a Model of Learning Systems", *Proceedings of the IEEE Transactions on Systems Science and Cybernetic*, vol. 5, no. 5, pp. 215–223, July 1969.

[15] A.B. Mingarelli, "A Classification Scheme of Fuzzy Cellular Automata with Applications to ECA", *Proceedings of the 13th International Workshop on Cellular Automata (Automata 2007)*, The Fields Institute, Toronto, Canada, August 27–29, 2007.

[16] K. Zhang, Z. Li, X.O. Zhao, "Edge Detection of Images based on Fuzzy Cellular Automata", *Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pp. 289–294, IEEE Computer Society Press, 2007.

[17] P. Maji, P.P. Chaudhuri, "RBFFCA: a Hybrid Pattern Classifier using Radial Basis Function and Fuzzy Cellular Automata", *Fundamenta Informaticae*, vol. 78, no. 3, pp. 369–396, 2007.

[18] E.A. Mandelas, T. Hatzichristos, P. Prastacos, "A Fuzzy Cellular Automata Based Shell for Modeling Urban Growth – A Pilot Application in Mesogia Area", *Proceedings of the 10th AGILE International Conference on Geographic Information Science*, Aalborg University, Denmark, 2007.

[19] P. Maji, "On Characterization of Attractor Basins of Fuzzy Multiple Attractor Cellular Automata", *Fundamenta Informaticae*, vol. 86, no. 1–2, pp. 143–168, October 2008.

[20] S. Basu, S. Basu, "Different Types of Linear Fuzzy Cellular Automata and their Applications", *Fundamenta Informaticae*, vol. 87, no. 2, pp. 185–205, November 2008.

[21] S. Yacoubi, A.B. Mingarelli, "Controlling the Dynamics of the Fuzzy Cellular Automaton Rule 90, I.", *Proceedings of the 8th International Conference on Cellular Automata for Reseach and Industry*, Yokohama, Japan, September 23–26, 2008, *Lecture Notes in Computer Science*, vol. 5191, pp. 174–183, 2008.

Curvas de Edwards y ataques basados en puntos de valor cero (ZVP)

S. Martínez
Dpto. Informatica i i Enginyeria Industrial.
U. Lleida.
Email: santi@diei.udl.es

D. Sadornil
Dpto. Matemáticas, Estadística y Computación.
U. Cantabria.
Email: sadornild@unican.es

J. Tena
Dpto. Álgebra, Geometría y Topología.
U. Valladolid.
Email: tena@agt.uva.es

R. Tomás, M. Valls
Dpto. de Matemàtica.
U. Lleida.
Email: {rosana,magda}@matematica.udl.cat

***Resumen*—El uso de las curvas elípticas en tarjetas inteligentes es vulnerable a los ataques denominados Zero Value Points (ZVP), puntos de valor cero. Tales ataques se suelen evitar empleando curvas resistentes a los mismos. Para ello, se pueden tomar curvas isógenas a la dada inicialmente hasta hallar una curva adecuada. En este artículo se plantea una alternativa a este método: el uso de curvas de Edwards. Demostramos que tales curvas son resistentes a los ataques ZVP.**

I. Introducción

El uso de las curvas elípticas en sistemas criptográficos se planteó por primera vez en los años 80 [10], [11]. Desde entonces, han adquirido relevancia dado que permiten ofrecer altos niveles de seguridad, pero utilizando claves de reducido tamaño. Por ello, tienen especial interés en el diseño de protocolos criptográficos sobre tarjetas inteligentes u otros dispositivos de capacidad computacional y de memoria limitadas.

De todos modos, su uso en tarjetas inteligentes no está exento de algún inconveniente: su vulnerabilidad a ciertos Side Channel Attacks (SCA) [3], [7], cuyo objetivo radica en obtener información de las claves almacenadas en las tarjetas a partir de la observación del comportamiento de las mismas, bien sea por consumo de tiempo al efectuar operaciones, o consumo energético, o incluso comportamiento electromagnético. Existen en la literatura múltiples trabajos en los que se analizan medidas de seguridad para resistir este tipo de ataques [3], [7].

Goubin [9] fue el primero en describir un ataque SCA específico para criptosistemas elípticos. En su trabajo, apuntaba ciertas características que deberían tener las curvas elípticas escogidas, para que fueran resistentes a estos ataques. Su trabajo fue posteriormente ampliado por Akishita and Takagi [1], [2], describiendo un ataque (Zero Value Point Attack, ZVP) que podrían sufrir aquellas curvas en los que existieran cierto tipo de puntos. Hasta el momento, las técnicas empleadas para resistir estos ataques se basan en el uso de curvas isógenas [2], [12], [16]: dada una curva que pueda contener puntos vulnerables, se busca una curva isógena a la misma, pero que cumpla las condiciones deseadas. La curva isógena presentará el mismo nivel de seguridad criptográfica que la original (dado que se mantiene el mismo cardinal), y además será resistente a los ataques ZVP.

Recientemente, H.M. Edwards ha estudiado un nuevo modelo para curvas elípticas, las denominadas curvas de Edwards [8], y que presentan una serie de propiedades interesantes [4], [5], [6], [13], [14]. Por un lado, se puede demostrar que cualquier curva elíptica sobre un cuerpo no binario es birracionalmente equivalente (un cambio de variable dado por funciones racionales) a una curva de Edwards sobre una extensión del cuerpo y, en muchos casos, sobre el mismo cuerpo. Además, las expresiones para la suma de puntos en estas curvas son mucho más simples y, en el caso que exista un único punto de orden 2, son completas y fuertemente unificadas (es decir, se pueden emplear para cualquier par de puntos de la curva, y además la misma expresión se puede utilizar para el doblado de puntos).

Posteriormente, D. Bernstein and T. Lange [5] han extendido la noción de curvas de Edwards, para abarcar una clase mayor de curvas sobre el cuerpo original. En su estudio, han comparado el comportamiento de las operaciones de grupo en estas curvas frente a otras formas y otros sistemas de coordenadas, concluyendo que las de Edwards ofrecen mayor eficiencia computacional. Además, han mostrado que las curvas de Edwards son también compatibles con medidas de prevención ante ataques SCA, como son la aleatorización de escalares, de coordenadas, de puntos o de curvas.

En el presente trabajo, nos interesamos por el comportamiento de las curvas de Edwards ante ataques ZVP. Mostramos que tales curvas satisfacen ciertas condiciones que las hacen resistentes a los ataques ZVP y, por consiguiente, utilizarlas resulta una medida preventiva eficiente ante estos ataques.

El artículo está estructurado como sigue: la Sección II es una introducción a las principales propiedades y resultados conocidos sobre curvas de Edwards; en la Sección III estudiamos las condiciones que deben satisfacer las curvas de Edwards para ser resistentes a ataques ZVP; finalmente, la Sección IV recoge las conclusiones del trabajo.

II. Curvas elípticas en forma de Edwards

En 2007, H.M. Edwards [8] definió un nuevo tipo de ecuaciones para algunas curvas elípticas; más concretamente, toda curva elíptica definida sobre un cuerpo de característica distinta de 2 admite una ecuación en forma de Edwards, es decir, de la forma $x^2+y^2=c^2(1+x^2y^2)$. La transformación birracional de paso de la forma original a la de Edwards puede realizarse en la clausura algebraica del cuerpo base.

Posteriormente, D. Bernstein y T. Lange [5], con el fin de que dicha transformación pudiera realizarse sobre el cuerpo base para una familia de curvas elípticas más amplia, extendieron esta idea admitiendo un nuevo tipo de ecuación, que por similitud también denominaremos de Edwards. Diremos que una curva E está en forma de Edwards si tiene por ecuación

$$x^2+y^2=c^2(1+dx^2y^2) \quad cd(1-c^4d)\neq 0.$$

Para una tal curva E se puede definir, de manera análoga a como se hace para una curva elíptica en forma de Weierstrass, una ley de composición. En ella, el elemento neutro es $(0,c)$ y dado un par de puntos racionales de la curva (x_1,y_1), $(x_2,y_2)\in E$ la suma se define como:

$$(x_1,y_1)+(x_2,y_2)=\left(\frac{x_1y_2+y_1x_2}{c(1+dx_1x_2y_1y_2)},\frac{y_1y_2-x_1x_2}{c(1-dx_1x_2y_1y_2)}\right)$$

Si d no es un residuo cuadrático, esta operación es completa y unificada, es decir, las mismas ecuaciones se pueden aplicar independientemente de los puntos que se desee sumar y sin hacer distinción para sumar dos puntos o calcular el doble de un punto (este hecho no ocurre para la suma de puntos en curvas elípticas en forma de Weierstrass).

Además, en estas curvas existen siempre dos puntos de orden 4, $(c,0)$ y su opuesto $(-c,0)$. De hecho, el hecho de poseer puntos de orden 4 es una condición necesaria para que una curva elíptica sea equivalente a una curva de Edwards. Concretamente, más del 25 % de las clases de isomorfía de curvas elípticas sobre un cuerpo finito pueden transformarse a una curva en forma de Edwards sobre el mismo cuerpo. El siguiente teorema [5] muestra qué curvas elípticas son birracionalmente equivalentes a una curva de Edwards.

Teorema 1. *Sea k un cuerpo de característica distinta de 2. Sea E una curva elíptica sobre k tal que el grupo $E(k)$ tiene un elemento de orden 4.*

1. *Existe $d\in k-\{0,1\}$ tal que la curva $x^2+y^2=1+dx^2y^2$ es birracionalmente equivalente sobre k a E o a una entrelazada cuadrática suya (twist cuadrática).*
2. *Si $E(k)$ tiene un único punto de orden 2, entonces existe un no cuadrado $d\in k$ tal que la curva $x^2+y^2=1+dx^2y^2$ es birracionalmente equivalente sobre k a E o a su twist cuadrática.*
3. *Si k es un cuerpo finito y $E(k)$ tiene un único punto de orden 2, entonces existe un no cuadrado $d\in k$ tal que la curva $x^2+y^2=1+dx^2y^2$ es birracionalmente equivalente sobre k a la curva original E.*

En el caso en que una curva elíptica sea birracionalmente equivalente a una curva de Edwards, existe una correspondencia entre la ley de grupo de la curva elíptica y la suma en la curva de Edwards (Teorema 3.2 en [5]).

Teorema 2. *Sea k un cuerpo de característica distinta de 2 y sean $c,d,e\in k^*$ con $e=1-dc^4$. Supongamos que d no es un cuadrado y sea la curva de Edwards $F: x^2+y^2=c^2(1+dx^2y^2)$. Sea E la curva elíptica de ecuación $(1/e)v^2=u^3+(4/e-2)u^2+u$. Para cada $i\in\{1,2,3\}$, sean (x_i,y_i) tres puntos de F tales que $(x_1,y_1)+(x_2,y_2)=(x_3,y_3)$. Se define P_i de la forma siguiente: $P_i=\mathcal{O}$ (punto del infinito) si $(x_i,y_i)=(0,c)$; $P_i=(0,0)$ si $(x_i,y_i)=(0,-c)$; y $P_i=(u_i,v_i)$ si $x_i\neq 0$, donde $u_i=(c+y_i)/(c-y_i)$ y $v_i=2c(c+y_i)/((c-y_i)x_i)$. Entonces*

$$P_i\in E(k) \quad y \quad P_1+P_2=P_3.$$

Por tanto, operaciones sobre una curva elíptica dada pueden trasladarse a operaciones sobre una curva de Edwards equivalente, donde las operaciones son computacionalmente mas eficientes. Finalmente, realizando la aplicación inversa, puede obtenerse de manera fácil el resultado en la curva elíptica original.

II-A. Cálculo de la suma y el doblado en una curva de Edwards

En su articulo [5], Bernstein y Lange determinan las fórmulas de las operaciones de grupo en una curva de Edwards. Dichas fórmulas, para evitar el cálculo de inversos modulares, se realizan usando coordenadas proyectivas $(X^2+Y^2)Z^2=c^2(Z^4+dX^2Y^2)$.

El punto $(X_1:Y_1:Z_1)$, $Z_1\neq 0$, corresponde al punto $(X_1/Z_1,Y_1/Z_1)$. El elemento neutro es $(0:c:1)$ y el opuesto de $(X_1:Y_1:Z_1)$ es $(-X_1:Y_1:Z_1)$.

Dados dos puntos $(X_1:Y_1:Z_1)$ y $(X_2:Y_2:Z_2)$, su suma $(X_3:Y_3:Z_3)$ puede obtenerse de la forma siguiente:

$$\begin{array}{llll} A=Z_1Z_2, & B=A^2, & C=X_1X_2, & D=Y_1Y_2,\\ E=dCD, & F=B-E, & G=B+E, & \end{array}$$

$$\begin{aligned} X_3&=AF((X_1+Y_1)(X_2+Y_2)-C-D),\\ Y_3&=AG(D-C),\\ Z_3&=cFG. \end{aligned}$$

Análogamente se calcula el doble de un punto, $(X_3:Y_3:Z_3)=2(X_1:Y_1:Z_1)$:

$$\begin{array}{lll} B=(X_1+Y_1)^2, & C=X_1^2, & D=Y_1^2,\\ E=C+D, & H=(cZ_1)^2, & J=E-2H, \end{array}$$

$$X_3=c(B-E)J,\ Y_3=cE(C-D),\ Z_3=EJ.$$

III. Ataques basados en Puntos de Valor Cero

Como se ha indicado en la introducción, el uso de criptografía con curvas elípticas en tarjetas inteligentes reveló un nuevo tipo de vulnerabilidad. Goubin [9] mostró que un atacante podría generar puntos en la curva de manera que, tras varios cálculos, alcanzase un punto con alguna coordenada nula. En esta situación, el atacante puede obtener información

de la clave secreta de la tarjeta e incluso todos los bits de la misma. Como contramedida, Smart [16] propuso buscar curvas elípticas isógenas a la original en las que no existan puntos con estas características. Entonces, los cálculos se realizan en la curva segura y, utilizando la isogenia dual, se trasladan los resultados a la curva original.

Posteriormente, Akishita y Takagi [1], [2] mostraron que este tipo de ataques se pueden extender, no sólo para puntos con coordenadas nulas, sino también para valores nulos de los registros auxiliares (i.e. los cálculos intermedios necesarios para obtener $2P$ o $P+Q$). Este tipo de ataque se denomina *Zero-Value Point attack* (ZVP) . Para una explicación más precisa véase [1].

De igual forma que en el ataque original, es posible encontrar curvas isógenas resistentes al ZVPA. Akishita y Takagi, en [2], determinan utilizando únicamente isogenias de grado primo, el grado mínimo de una isogenia para encontrar una curva isógena segura para las curvas del SECG [15]. En [12], los autores utilizan volcanes de isogenias para obtener curvas idóneas empleando isogenias de grado mas bajo.

Como contramedida alternativa, nos planteamos si las curvas de Edwards también pueden usarse para evitar ataques ZVP. Por ese motivo, estudiamos cuando los parámetros intermedios pueden anularse durante los procedimientos de suma y doblado. Concluimos que eso sólo sucede para puntos que no son criptográficamente interesantes, por lo que usar curvas de Edwards en tarjetas inteligentes resulta ser seguro contra este tipo de ataques.

Nótese que tomar curvas de Edwards (cuando esto sea posible) será una contramedida más eficiente que buscar curvas isógenas ya que este proceso se puede realizar de forma más rápida. Por desgracia, el inconveniente es que no todas las curvas elípticas tienen una curva equivalente en la forma de Edwards.

III-A. Posibles puntos de valor cero del doblado

A partir de las fórmulas dadas en la sección anterior para el doblado de un punto en una curva de Edwards, el siguiente resultado determina las condiciones para que un punto sea un punto de valor cero para el doblado.

Teorema 3. *Sea $x^2+y^2=c^2(1+dx^2y^2)$ una curva de Edwards sobre un cuerpo finito con d un no residuo cuadrático. Un punto P es un* punto de valor cero *para el doblado si y sólo si P es el elemento neutro de la suma, o es un punto de orden 2, 4 u 8.*

Demostración 1. *Usando coordenadas proyectivas, sea $P=(X_1:Y_1:Z_1)$ el punto a doblar. Los parámetros intermedios pueden ser cero si y sólo si alguno de los siguientes valores (de las expresiones del doblado) se anulan:*

$$(X_1+Y_1)^2, X_1^2, Y_1^2, X_1^2+Y_1^2, cZ_1, E-2H,\ B-E,\ C-D$$

Nótese que $Z_1\neq 0$ puesto que estamos tratando con puntos afines. Si $X_1=0$, el punto P sería o bien el elemento neutro $(0,c)$ o el punto de orden dos $(0,-c)$. En el caso de que $Y_1=0$, el punto es uno de los dos puntos de orden 4 de la curva: $(c,0)$ o su opuesto $(-c,0)$.

La condición $B-E=0$ es equivalente a $0=(X_1+Y_1)^2-X_1^2-Y_1^2=2X_1Y_1$, lo que se reduce a los casos anteriores.

Si se cumple $E-2H=0$, tomando coordenadas afines, se tiene que $x_1^2+y_1^2=2c^2$; entonces, como el punto pertenece a la curva, se concluye que $1=dx_1^2y_1^2$, lo que contradice el hecho de que d es un no cuadrado. Similarmente, se alcanzaría la misma contradicción si se considerase $X_1^2+Y_1^2=0$.

El caso más interesante ocurre cuando $(X_1+Y_1)^2$ o $C-D=X_1^2-Y_1^2$ son cero. En esa situación, el punto afín es o bien $(x_1,-x_1)$ o (x_1,x_1). Puesto que $2(x_1,-x_1)=(-c,0)$ y $2(x_1,x_1)=(c,0)$, éstos son puntos de orden 8.

Nótese que existen puntos de orden 8 en la curva si y sólo si $1-c^4d$ es un residuo cuadrático (esta condición se concluye fácilmente al verificar la existencia de puntos $(x_1,-x_1)$ o (x_1,x_1) en la curva).

III-B. Posibles puntos de valor cero de la suma

De modo similar al caso del doblado, el resultado siguiente determina las condiciones necesarias y suficientes para que un punto P sea un punto de valor cero al calcular la suma $P+kP$.

Teorema 4. *Sea $x^2+y^2=c^2(1+dx^2y^2)$ una curva de Edwards sobre un cuerpo finito, con d un no residuo cuadrático. Un punto P es un* punto de valor cero *para la suma de P y $kP=(x_2,y_2)$ si y sólo si P es el elemento neutro para la suma, o es un punto de orden 2, 4 u 8 o su orden es un divisor de alguno de los números enteros $\{k,k+1,2k,2(k+1),4k,4(k+1),8k\}$.*

Demostración 2. *Utilizando coordenadas proyectivas, sea $P=(X_1:Y_1:Z_1)$ el punto sumado a $kP=(X_2:Y_2:Z_2)$. Los parámetros intermedios pueden ser cero si y sólo si alguno de los valores siguientes (de las expresiones de la suma) se anula:*

$$Z_1Z_2, X_1X_2, Y_1Y_2, B-E, B+E, (X_1+Y_1)(X_2+Y_2), Y_3, X_3,$$

donde $(k+1)P=(X_3:Y_3:Z_3)$.

Puesto que estamos sumando puntos afines, Z_1Z_2 necesariamente será no nulo. Si $X_1X_2=0$, entonces X_1 o X_2 son cero. En el primer caso, P es elemento neutro de la suma, o es el punto de orden 2 de la curva. En el segundo caso, o bien $kP=(0,c)$ o bien $2kP=2(0,-c)=(0,c)$. Entonces el orden de P divide a k o a $2k$. El caso $Y_1Y_2=0$ se puede tratar de forma similar, y se obtiene que P es de orden 4 o su orden es un divisor de $4k$.

Nótese que $B-E$ y $B+E$ nunca pueden ser 0, en caso contrario, tomando coordenadas afines, debería pasar que $dx_1x_2y_1y_2\in\{-1,1\}$, lo que contradice la propiedad de completitud de la suma sobre la curva de Edwards (Teorema 3.3 en [5]).

Si $(X_1+Y_1)(X_2+Y_2)=0$, procediendo de forma similar que en la demostración del teorema anterior, se obtiene que P o kP tienen orden 8. En el último caso, el orden de P es un divisor de $8k$.

Si $Y_3 = 0$, o bien nos encontramos en uno de los casos anteriores o bien $D - C = 0$. En esta situación, se obtiene que el orden de P es un divisor de $4(k+1)$, ya que $(k+1)P$ tiene orden 4.

Finalmente, si $X_3 = 0$, entonces o $(k+1)P = (0, c)$ o $(k+1)P = (0, -c)$, así que el orden de P es un divisor de $k+1$ o de $2(k+1)$.

Nótese que la última condición en el Teorema 4 es equivalente al siguiente lema, donde r es el orden del punto P.

Lema 1. *Sea r un primo > 2 y k un entero no negativo menor que r. Entonces r es un divisor de alguno de los enteros $\{k, k+1, 2k, 2(k+1), 4k, 4(k+1), 8k\}$ si y sólo si $k = 0, r-1$.*

Finalmente, a partir de los teoremas 3 y 4, podemos concluir que:

Corolario 1. *Las curvas de Edwards son adecuadas para implementarse en tarjetas inteligentes que usen criptografía de curvas elípticas, porque son resistentes a los ataques ZVP.*

Demostración 3. *En criptografía con curvas elípticas, se calculan múltiplos de un punto (mP), donde m es un parámetro grande. Cuando se implementa en tarjetas inteligentes, es necesario garantizar que no aparecerán* puntos de valor cero *durante el cómputo de mP (lo cual se realiza por medio del algoritmo de suma y doblado).*

Por motivos de seguridad (para garantizar que el logaritmo discreto elíptico no se pueda resolver fácilmente), el punto P tiene orden primo r (de hecho r se toma como el primo más grande que divide al cardinal de la curva). De ahí, que el orden de P nunca será divisor de 8. Además, con respecto a las condiciones indicadas en el lema anterior, nótese que el caso $k = 0$ no aparecerá durante el cómputo de mP. Lo mismo pasa para la segunda condición porque, en la práctica, m será más pequeño que r.

Así, durante el cálculo de mP, el procedimiento nunca se encontrará con un punto de valor cero*, ni en la suma, ni en el doblado.*

IV. Conclusion

Se ha mostrado la idoneidad de las curvas de Edwards como base para implementar protocolos criptográficos en plataformas con memoria y capacidad de cálculo reducidas, como las tarjetas inteligentes, los dispositivos de identificación por radiofrecuencia (RFID) o las redes de sensores.

Dotar a estas plataformas con capacidades criptográficas requiere soluciones específicas, que se adapten a tales restricciones computacionales (lightweight cryptography).

Las curvas de Edwards proporcionan un modelo alternativo de curvas elípticas y los resultados de Bernstein y Lange muestran que su aritmética (suma y doblado de puntos) es computacionalmente más eficiente que la de los modelos tradicionales, lo que avala la propuesta de su uso en los dispositivos mencionados, con recursos computacionales restringidos.

Una vulnerabilidad, encontrada en el uso de la criptografía elíptica en tarjetas inteligentes, es la posible existencia de puntos de la curva con alguna coordenada nula, o más generalmente de expresiones que se anulan en los cálculos intermedios necesarios para obtener la suma de dos puntos o el doble de un punto, circunstancia que puede ser utilizada por un adversario para obtener la clave privada de la tarjeta (Zero Value Point Attacks). En el presente trabajo se ha estudiado la resistencia a los ZVPA de los criptosistemas basados en curvas de Edwards, demostrando su inmunidad a los mismos. En efecto, se ha probado que tales curvas poseen pocos puntos vulnerables, los cuales además no son criptográficamente útiles y por tanto, por razones de seguridad, nunca serían tomados en consideración para uso criptográfico.

En consecuencia, el uso de curvas de Edwards representa una buena alternativa, en la implementación de la criptografía elíptica en tarjetas inteligentes, a las actuales medidas defensivas contra los ZVPA (la búsqueda mencionada, vía isogenias, de curvas elípticas resistentes). Esta estrategia es incluso adaptable a protocolos utilizando modelos 'clásicos' de curvas elípticas: supuesto un protocolo basado en una tal curva elíptica y siempre que esta admita una curva de Edwards equivalente (en particular posea puntos de orden cuatro) las operaciones criptográficas necesarias pueden realizarse en la curva de Edwards asociada (cuya aritmética es además más eficiente) y el resultado obtenido ser finalmente trasladado a la curva original.

Agradecimientos

Este trabajo está subvencionado por los proyectos MTM2007-66842-C02-01/02 y MTM2007-62799.

Referencias

[1] T. Akishita, T. Takagi. Zero-Value Point attacks on elliptic curve cryptosystem. *Information Security, ISC 2003*, LNCS 2851, 218–233. 2003.

[2] T. Akishita, T. Takagi. On the optimal parameter choice for elliptic curve cryptosystems using isogeny. *Public Key Cryptography, PKC 2004*, LNCS 2947, pp. 346-359, 2004.

[3] R. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, F. Vercauteren. *Handbook of elliptic and hyperelliptic curve cryptography*. Discrete Math. Appl. Chapman & Hall/CRC, 2006.

[4] D. J. Bernstein, P. Birkner, T. Lange, C. Peters. ECM using Edwards curves, *Cryptology ePrint Archive*, Report 2008/016, 2008.

[5] D. J. Bernstein, T. Lange. Faster addition and doubling on elliptic curves. *ASIACRYPT 2007*. LNCS 4833, 29–50, Springer, 2007.

[6] B. Baldwin, A. Byrne, G. McGuire, R. Moloney, W. P. Marnane. A hardware analysis of twisted Edwards curves for an elliptic curve cryptosystem. *Reconfigurable Computing: Architectures, Tools and Applications, ARC 2009*, LNCS 5453, 355–361, Springer 2009.

[7] I. F. Blake, G. Seroussi, N. P. Smart. *Advances in elliptic curve cryptography*. London Math. Soc. Lecture Note Ser. vol. 265, Cambridge Univ. Press, Cambridge, 1999.

[8] H. M. Edwards. A normal form for elliptic curves. *Bulletin of the American Mathematical Society* 44, 393-422. 2007

[9] L. Goubin. A refined power-analysis attack on elliptic curve cryptosystems. *Public Key Cryptography, PKC 2003*, LNCS 2567, pp. 199-211, 2003.

[10] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation* 48, 177, 203–209. 1987.

[11] V. Miller. Use of elliptic curves in cryptography. *CRYPTO 85*, 417–426, LNCS 218, Springer, Berlin, 1986.

[12] J. Miret, D. Sadornil, J. Tena, R. Tomàs, M. Valls. On avoiding ZVP–attacks using isogeny volcanoes. *Workshop on Information Security Applications, WISA 2008*, Springer LNCS 5379, 266-277, 2009.

[13] F. Morain. Edwards curves and CM curves. arXiv:0904.2243, 2009.

[14] M.P. Das and P. Sarkar. Pairing computation on twisted Edwards form elliptic curves. *Pairing-Based Cryptography, Pairing 2008*. Springer LNCS 5209, 192–210, 2008.
[15] Standard for Efficient Cryptography (SECG). *Recommended Elliptic Curve Domain Parameters,* 2000. http://www.secg.org/secg_docs.htm.
[16] N. Smart. An analysis of Goubin's refined power analysis attack. *Workshop on Cryptographic Hardware and Embedded Systems, CHES 2003*, LNCS 2779, pp. 281-290, 2003.

Grafos de Cayley como bases de protocolos de identificación

Feliú Sagols
Departamento de Matemáticas, Cinvestav-IPN
México, D. F., México
CorreoE: fsagols@math.cinvestav.mx

Guillermo Morales-Luna
Departamento de Computación, Cinvestav-IPN
México, D. F., México
CorreoE: gmorales@cs.cinvestav.mx

***Resumen*—Debido a la alta dificultad en resolver el *Problema de la Palabra* en ciertos grupos de Coxeter, es posible utilizar a los respectivos grafos de Cayley para desarrollar protocolos de identificación del tipo *reto–respuesta*. Un usuario, digamos *probador*, busca, en efecto, probar ante un *verificador* que él es el titular de una cierta identidad digital. Considerando el grafo de Cayley de un grupo de Coxeter en el que el problema de la palabra sea intratable, el probador construye su clave pública como el conjunto de hojas de un árbol y éste es en sí una correspondiente clave privada. En un primer protocolo de autenticación, el verificador elige como retos subconjuntos de la clave pública y el probador presenta como respuestas a los subárboles de una clave privada que tienen a los retos como conjuntos de hojas. Cualquier tercera entidad que busque suplantar al probador ha de enfrentarse al problema de la palabra en el grupo elegido. Aunque este protocolo mantiene a la totalidad de la clave privada en secreto, muestra una parte de ella al verificador. Presentamos también un segundo protocolo, éste de conocimiento nulo, consistente de una transcripción al presente contexto del célebre protocolo de este tipo para reconocer parejas isomorfas de grafos.**

***Términos de índice*—Grupos de Coxeter, procedimientos de identificación, selección aleatoria de árboles generadores.**

I. Introducción

Diversos procedimientos del tipo *reto–respuesta* han sido presentados en la literatura con el fin de autenticar e identificar usuarios. Un usuario, o *probador*, que ha de mostrar ante un *verificador* que posee las credenciales que lo acreditan como titular de una identidad digital, puede generar parejas de instancias y soluciones de problemas computacionales difíciles. Las instancias pueden ser asumidas como claves públicas y las correspondientes soluciones como claves privadas [1]. El verificador elige instancias del problema, el probador le proporciona las correspondientes soluciones y este diálogo se repite hasta la plena satisfacción, o convencimiento, del verificador. Una tercera parte que busque suplantar al probador ha de enfrentarse con el difícil problema de computar las soluciones de las instancias planteadas como retos. Entre los protocolos de reto–respuesta están los de *conocimiento nulo* [2], los cuales evitan que el verificador "aprenda más de lo estrictamente necesario" para convencerse de que el probador posee, en efecto, una clave privada correspondiente a su propia clave pública.

Presentamos aquí, primeramente, un protocolo de identificación del tipo reto–respuesta basado en la dificultad de resolver el problema de la palabra en grupos de Coxeter. En la sección II describimos el protocolo de manera general refiriéndonos a un grafo en el que el problema de localizar caminos entre parejas arbitrarias de nodos sea intratable (para esto es necesario que el número de nodos sea de crecimiento superpolinómico respecto a un parámetro de control).

El problema de la palabra en grupos finitos es, naturalmente, resoluble y su complejidad es, en general, polinómica respecto al orden del grupo. En la sección III bosquejamos la construcción de los grupos de Coxeter cuyos órdenes son superpolinómicos respecto al número de generadores y, en consecuencia, sus grafos de Cayley son apropiados para realizar el protocolo de identificación introducido.

Ya que durante el protocolo el probador está presentando soluciones al problema de la palabra correspondientes a porciones propias de su clave privada, dicho protocolo no puede ser de conocimiento nulo. En la sección IV presentamos un protocolo de conocimiento nulo que permite verificar que el probador conoce las respuestas del problema de la palabra entre cualesquiera dos nodos en su clave pública sin tener que desvelar las soluciones correspondientes.

A manera de un brevísimo recuento de la influencia de la Teoría de Grafos en Criptografía recordamos aquí que los *grafos expansores* han sido utilizados para reforzar funciones de *hash* [3], ha habido esquemas de compartición de secretos usando grafos (el tamaño de los fragmentos es proporcional a la valencia de los nodos) [4], así como esquemas de distribución de claves [5]. Una mención especial la merece el uso que se ha hecho de grafos para protocolos de cifrado [6]: los nodos de ciertos grafos son mensajes y los cifrados son caminos que conectan nodos.

II. Protocolo de identificación

A. Grafos de crecimiento rápido

Sea $K \neq \emptyset$ un conjunto de cardinal $k \in \mathbb{Z}^+$. Sea $f : \mathbb{N} \to \mathbb{N}$ una función de crecimiento al menos lineal, $f(n) = \Omega(n)$. Para un entero $n \in \mathbb{N}$, $K^{f(n)}$ es el conjunto de palabras con símbolos en K de longitud $f(n)$. Su cardinal es $N_{kn} = k^{f(n)}$.

Sea $\mathcal{G} = (K^{f(n)}, A)$ un grafo sobre $K^{f(n)}$. Entonces

$$2\,\mathrm{card}(A) = \sum_{\mathbf{x} \in K^{f(n)}} \mathrm{val}(\mathbf{x}).$$

Por tanto, si la valencia de cada nodo fuera al menos $d \in \mathbb{Z}^+$ se tendría $\text{card}(A) \leq \frac{d}{2} N_{kn}$ y la altura de cualquier árbol en un bosque generador de $\mathcal{G}$ es de orden $O(\log N_{kn}) = O(f(n) \log k)$. En tal caso el problema de localizar caminos entre pares de nodos en $\mathcal{G}$ puede ser intratable respecto a n. Los algoritmos de Dijkstra (en grafos sin ponderación), de Prim y de Kruskal (para localizar árboles generadores), por ejemplo, que permitirían establecer caminos entre nodos, tendrían una complejidad temporal del orden

$$O(\text{card}(A) + N_{kn} \log N_{kn}) = O\left((d + f(n))\, k^{f(n)}\right),$$

es decir, exponencial, y por tanto intratable, respecto a n.

B. Protocolo de identificación

Consideremos el siguiente *escenario de identificación*: Un probador procura convencer a un verificador de que es el titular de la identidad sintetizada, digamos, en una *partícula de identidad*. El verificador le plantea diferentes preguntas como retos y, en función de las respuestas que obtiene, decide si acaso el probador es el dueño de esa partícula de identidad.

La partícula de identidad es en sí una clave pública del probador y el conocimiento no desvelado que acredita la titularidad de la identidad es la clave privada.

En un grafo de crecimiento rápido, sobre $K^{f(n)}$ con $f(n) = \Omega(n)$, sea $\mathcal{G}$ un subgrafo en donde el problema de localizar caminos entre pares de nodos sea intratable, respecto a n. Dado un *árbol* T, es decir un subgrafo conexo sin ciclos, en $\mathcal{G}$, sea $H(T)$ el conjunto de sus *hojas*, es decir de los nodos en T de valencia 1. Evidentemente, el problema de localizar caminos entre hojas es trivial para quien conoce el árbol, pero puede ser intratable para quien lo desconozca, pues ese problema lo es en $\mathcal{G}$.

Así, el probador construye un árbol en la gráfica $\mathcal{G}$ y publica como su clave pública al conjunto de hojas. Cualquier participante que quiera verificar que el probador conoce la correspondiente clave privada elige un subconjunto de hojas en la clave pública y le requiere un árbol de $\mathcal{G}$ que tenga a ese subconjunto como hojas. Evidentemente, el probador podrá cumplir de manera eficiente con el requerimiento pues la respuesta es un subárbol de su clave privada.

Protocolo de clave pública

Precondición. El grafo $\mathcal{G}$ debe ser conocido por los participantes en el esquema.

Inicio. El probador escoge un árbol T como un subgrafo de $\mathcal{G}$. El árbol T es la clave privada del probador. El conjunto $H(T)$ de hojas de T es su clave pública.

Protocolo de identificación. Repítase

1) el verificador escoge un conjunto propio N de $H(T)$ y lo presenta al probador como reto,
2) el probador calcula el subárbol T_N de T, tal que $H(T_N) = N$, y le envía T_N al verificador,
3) el verificador recibe T_N y revisa que en efecto $H(T_N) = N$. Si no se cumpliese esta identidad, el verificador rechaza el protocolo,

hasta el pleno convencimiento del verificador.

Debido a que el probador conoce T, podrá responder exitosamente en cualquier ronda. Sin embargo, para mantener la robustez del protocolo, varias condiciones han de imponerse:

- el reto N debe plantear una instancia intratable del problema de localizar árboles generadores en $\mathcal{G}$, y
- el reto N, o la unión de los retos en las rondas del protocolo, debe ser un conjunto propio de $H(T)$: de otra forma el probador debería desvelar su clave privada.

Por supuesto, cualquier agente que construya un árbol T_1 tal que $H(T_1)$ coincida con la clave pública del probador podría superar exitosamente el protocolo de identificación. La construcción de un tal árbol puede resolverse en tiempo lineal respecto al número $N_{\mathcal{G}}$ de vértices en el grafo $\mathcal{G}$, pero es un procedimiento muy costoso en términos de $\ell_{\mathcal{G}} = O(\log N_{\mathcal{G}})$ que es, a su vez, proporcional con el diámetro del grafo $\mathcal{G}$.

Aún cuando un intruso que escuche el diálogo entre el probador y el verificador conocerá parcialmente la clave privada del probador, no podría contestar eficientemente de manera correcta en una nueva ronda del protocolo debido a la dificultad en localizar caminos en $\mathcal{G}$.

III. El problema de la palabra y la robustez del protocolo

Recordamos que una *presentación* de un grupo G es una pareja (C, R), donde C es una colección de *generadores* y $R \subset \{\alpha = \beta |\ \alpha, \beta \in C^*\}$ es una colección de *relaciones* tal que $G \approx F(C)/\langle\langle R \rangle\rangle$, donde $F(C)$ es el grupo libre no-abeliano generado por C y $\langle\langle R \rangle\rangle$ es el subgrupo normal de $F(C)$ generado por los *relatores*, es decir, las palabras $\alpha\beta^{-1}$, con $(\alpha = \beta) \in R$. La presentación es *finita* si tanto C como R lo son. Naturalmente, todo grupo finito posee una presentación finita, pero el recíproco no se cumple.

Por ejemplo para un entero positivo $n \in \mathbb{Z}^+$, una presentación del *grupo simétrico* S_n, consistente de las permutaciones en el conjunto $[\![0, n-1]\!] = \{0, 1, \ldots, n-1\}$ de n índices, se obtiene con $n-1$ *generadores* $(t_{in})_{i=0}^{n-2}$ y $(n-1)^2$ *relatores* $((t_{in} t_{jn})^{n_{ij}})_{i,j=0}^{n-2}$, donde

$$n_{ij} = \begin{cases} 1 & \text{si } |j-i| = 0, \\ 3 & \text{si } |j-i| = 1, \\ 2 & \text{si } |j-i| > 2. \end{cases}$$

De hecho, cada generador t_i ha de corresponder a la transposición $(i, i+1)$ de dos índices consecutivos. Así, el grupo simétrico S_n de orden $n!$ posee una presentación con $(n-1)$ generadores y $(n-1)^2$ relatores.

Sea pues G un grupo dado mediante una presentación (C, R). El *Problema de la Palabra* en G, puesto como un problema de decisión, consiste en decidir para una palabra $\sigma \in (C \cup C^{-1})^*$ si acaso está o no en el subgrupo normal $\langle\langle R \rangle\rangle$ generado por R en el grupo libre $F(C)$. O, formulado como un problema de búsqueda, consiste en que dado $g \in G$ se ha de localizar una palabra σ, acaso mínima de acuerdo con un cierto orden "bien fundamentado" de $(C \cup C^{-1})^*$, tal que $g = \sigma$ en G.

Es bien sabido, desde la década de los 50 del siglo pasado, que existen grupos (infinitos) con presentaciones finitas en donde el problema de la palabra es irresoluble algorítmicamente (Teorema de Novikov). El grafo de Cayley de un tal grupo puede servir para implementar el algoritmo de identificación presentado en la sección anterior. Sin embargo, tal grupo, al ser infinito, es demasiado grande para especificar de manera eficiente, desde el punto de vista computacional, el subgrafo $\mathcal{G}$ que aparece en el protocolo.

Por otra parte, aunque el problema de la palabra es siempre resoluble algorítmicamente en grupos finitos, en la práctica puede ser intratable respecto al número n de generadores, como es el caso de los llamados *grupos de Artin* [7]. Recordemos brevemente su construcción.

Para dos símbolos x, y y un entero $\ell \in \mathbb{Z}^+$, se denota por $(xy)^{[\ell]}$ al prefijo de longitud ℓ de la palabra $(xy)^\ell$, es decir $(xy)^{[\ell]} = (xy)^{\lfloor \frac{\ell}{2} \rfloor}\eta$ donde $\eta = x$ si ℓ es impar y η es la palabra vacía en otro caso.

Sea C un conjunto finito de generadores, es decir meros símbolos, y $M \in (\mathbb{N} \cup \{+\infty\})^{C \times C}$ una matriz de orden $C \times C$ con entradas en $\mathbb{N} \cup \{+\infty\}$. Las entradas de M son pues enteros no-negativos o el valor infinito. Sea R_M la colección de relaciones $(xy)^{[m_{xy}]} = (yx)^{[m_{yx}]}$, con $\{x, y\} \in C^{(2)}$. Entonces (C, R_M) es el grupo de Artin determinado por la matriz M. El *grupo de Coxeter* correspondiente es el que se obtiene al añadir las relaciones $x^2 = 1$ (i.e. $m_{xx} = 2$).

El problema de la palabra en grupos de Coxeter tiene una complejidad temporal exponencial con respecto al número de generadores, $n = \text{card}(C)$, y ha servido como base de diversos sistemas criptográficos de clave pública, v.g.: [7].

Sea $G(n, M)$ un grupo de Coxeter determinado por la matriz M. Posee n generadores y $d = \frac{1}{2}(n+1)n = O(n^2)$ relatores. Sea $f(n) = \lceil \log_n(o(G(n, M))) \rceil$ el logaritmo en base n del orden del grupo, entonces $f(n) = \Omega(n)$. Asociando a cada elemento en el grupo su expresión mínima como una palabra sobre el alfabeto de los generadores, es posible representar al grafo de Cayley del grupo $G(n, M)$ con nodos en $K^{f(n)}$, con $K = C$, tal como se requería en la sección anterior al presentar el protocolo de identificación.

De manera alternativa, sea $H_M = \langle\langle R_M \rangle\rangle$ el subgrupo normal generado por R_M en el grupo libre $F(C)$ generado a su vez por C. Sea $\mathcal{G}_M$ el grafo cuyo conjunto de nodos es H_M y las aristas son parejas de la forma $(\sigma\alpha\tau, \sigma\beta\tau)$ en donde bien $(\alpha = \beta) \in R_M$ o bien $(\beta = \alpha) \in R_M$, es decir las reglas de reducción pueden ser aplicadas en uno y en otro sentido. El problema de la palabra, que es equivalente al de encontrar caminos entre cualesquiera dos nodos, sigue siendo de complejidad temporal exponencial respecto a n.

Cualquier vértice en este grafo se expresa como una palabra de longitud a lo sumo $f(n)$ sobre el alfabeto C, por ende puede ser representado por una cadena de bits de longitud $O(f(n) \log n)$.

Para utilizar el grafo $\mathcal{G}_M$ en el protocolo de identificación, un probador ha de construir un subgrafo adecuado $\mathcal{G}_m$ de $\mathcal{G}_M$ y un árbol generador de $\mathcal{G}_m$, donde $m \in \mathbb{N}$ es propiamente un parámetro del protocolo.

Supongamos por el momento que ya se haya construido el grafo $\mathcal{G}_m = (V_r, A)$ con $r = r(m) \in \mathbb{Z}^+$ nodos. Para cada nodo $v \in V_r$ sea d_v su valencia en $\mathcal{G}_m$. Se va a construir un árbol generador siguiendo el procedimiento ya clásico presentado originalmente por Broder [8] y por Aldous [9] de manera independiente.

Para cualesquiera dos vértices $u, v \in V_r$, sea p_{uv} igual a d_v^{-1} si $\{v, u\} \in A$ e igual a 0 en otro caso. Se tiene que $P = (p_{uv})_{u,v \in V_r}$ es la matriz de transición de una cadena simple de Markov en el grafo $\mathcal{G}_m$. Un árbol generador seleccionado uniformemente se obtiene de acuerdo con el procedimiento siguiente [8]:

`ÁrbolGeneradorUniforme`

1) A partir de un nodo inicial seleccionado aleatoriamente $v_0 \in V_r$, sea $\{x_\tau\}_{\tau=0}^{t_e}$ un paseo aleatorio de una mínima longitud que recorra el grafo $\mathcal{G}_m$. Así, para cada $v \in V_r$ existe un instante mínimo $t_v \leq t_e$ tal que $x_{t_v} = v$.
2) Sea T el árbol cuyas aristas son las parejas $\{x_{t_v - 1}, v\}$, con $v \in V - \{v_0\}$.

T es un árbol generador pues consta exactamente de r nodos y $r - 1$ aristas. Se tiene [8] que $t_e = O(r^3)$ en peores casos en tanto que, por lo general, puede esperarse $t_e = O(r \log r)$. También, bajo ciertas condiciones de regularidad del grafo $\mathcal{G}_m$, se puede esperar [9] que la razón entre el número de hojas en él y el de vértices queda acotada superiormente por $\exp\left(-\frac{r-1}{2r}\right)$, así como que el diámetro $\Delta(T)$ del árbol es $O(\sqrt{r})$.

En consecuencia, dado $m \in \mathbb{N}$, el probador puede generar un árbol, con un número esperado de hojas m, como un subgrafo de $\mathcal{G}_M$ seleccionando primeramente un grafo conexo de $r(m) = \lceil \sqrt{e}\ m \rceil$ vértices y luego seleccionando uniformemente en él un árbol generador:

`ÁrbolGeneradorConNúmeroEsperadoDeHojas`

1) Selecciónese un subgrafo $\mathcal{G}_m$ de $\mathcal{G}_M$, con $r = \lceil \sqrt{e}\ m \rceil$ vértices.
2) Mediante `ÁrbolGeneradorUniforme` obténgase un árbol generador T de $\mathcal{G}_m$.
3) Dése el árbol T como resultado.

En el paso 1) de este algoritmo se podría instrumentar un recorrido del tipo "primero-a-lo-ancho" para procurar seleccionar un mismo número de vecinos en cada nodo recién "descubierto", con el fin de satisfacer las condiciones de regularidad en [9] para esperar tener efectivamente alrededor de m hojas en el árbol producido T. Este árbol y su conjunto de hojas podrán entonces representarse por cadenas de bits de longitud $O(m\, f(n) \log n)$. Así pues, ésta es la longitud de los mensajes intercambiados en las rondas del protocolo de identificación.

Es pertinente observar aquí que en la anterior construcción podría omitirse elaborar el grafo $\mathcal{G}_m$ mediante el procedimiento `ÁrbolGeneradorConNúmeroEsperadoDeHojas`. A saber: para cada nodo v del grafo de Cayley $\mathcal{G}_M$ considérese una distribución de probabilidad $(p_{uv} |\ \{v, u\}$ es una arista en $\mathcal{G}_M)$, con lo cual $\mathcal{G}_M$ viene a

ser una cadena simple de Markov. Dado el entero $m \in \mathbb{N}$, el árbol T se produce como en `ÁrbolGeneradorUniforme` mediante un paseo aleatorio que se suspende tan pronto se haya visitado $\lceil \sqrt{e}\ m \rceil$ nodos distintos a pares en $\mathcal{G}_M$.

IV. Protocolo de conocimiento nulo

Sea C un conjunto finito de generadores y sea $M \in (\mathbb{N} \cup \{+\infty\})^{C \times C}$. Sea $\mathcal{G}_M$ la gráfica de Cayley del grupo de Coxeter determinado por M. De acuerdo con los procedimientos en la sección III, un probador construye un árbol T en $\mathcal{G}_M$, lo asume como su clave privada, y anuncia el conjunto de hojas $H(T)$ como su clave pública. Para autenticar al probador, un verificador toma un subconjunto J de $H(T)$ y se lo presenta como reto al probador quien responde con el mínimo subárbol S de T que tiene a J como conjunto de hojas. Así el verificador conoce una parte de la clave privada del probador, sin embargo esto no le permite al verificador conocer la clave privada por completo. Las rondas reto-respuesta podrían reiterarse en tanto los retos no cubran el total de la clave pública el probador. Si acaso se recubriese ésta, ya sea mediante repeticiones sucesivas de rondas reto-respuesta o mediante la colusión de varios verificadores, entonces se quebrantaría el esquema de autenticación.

Presentemos un protocolo modificado, éste de conocimiento nulo, similar al que reconoce parejas de grafos isomorfos [2]:

El probador tiene clave pública $H(T)$ y clave privada T. El verificador conoce pues $H(T)$.

`IdentificaciónConConocimientoNulo`

Repítase

1) el verificador escoge dos nodos $v_0, v_1 \in H(T)$ y se los envía al probador,
2) el probador encuentra el camino h que va de v_0 a v_1 en T, y elige aleatoriamente un punto intermedio v_2 sobre h. Sea h_0 el tramo de h que conecta a v_0 con v_2 y sea h_1 el tramo de h que conecta a v_1 con v_2. Elige un camino g en el grafo de Cayley que se inicie en v_2. Le envía al verificador el nodo final u de ese camino,
3) el verificador escoge un bit $b \in \{0, 1\}$ y se lo envía al probador (con la intención de que éste le muestre el camino que conecta al nodo v_b con u),
4) el probador responde con $f = g \star (h_b)$ [aquí $\star$ es la concatenación],
5) el verificador comprueba que, en efecto, f conecta a v_b con u,

hasta que falle el probador o el verificador quede convencido.

Con este protocolo ningún verificador recibe información parcial de la clave privada del probador.

V. Conclusiones

El primer protocolo de identificación presentado es robusto debido al rápido crecimiento de los grafos involucrados haciendo intratables los problemas de la palabra y de localización de caminos en ellos. Así, las claves públicas no tienen que ser demasiado largas, e incluso podrían consistir de un mero par de vértices suficientemente alejados pues el problema de conectarlos es bastante difícil. Sin embargo, el número de rondas en el protocolo queda restringido por el número de hojas que formen una clave pública. Si el conjunto de retos recubre la clave pública, el probador habrá de desvelar su propia clave privada. Obviamente, en tal caso cualquier intruso que la haya capturado podrá cumplir exitosamente con el protocolo propuesto. El interés de este protocolo radica en el uso de la dificultad del problema de la palabra en gráficas de Cayley. El segundo protocolo presentado es de conocimiento nulo y sólo calca procedimientos canónicos en este tema para proponer un mecanismo de identificación mediante grafos de Cayley.

Referencias

[1] R. J. Hartung and C.-P. Schnorr, "Public key identification based on the equivalence of quadratic forms," in *MFCS*, ser. Lecture Notes in Computer Science, L. Kucera and A. Kucera, Eds., vol. 4708. Springer, 2007, pp. 333–345.

[2] O. Goldreich, *Computational Complexity: A Conceptual Perspective*. New York, NY, USA: Cambridge University Press, 2008.

[3] O. Goldreich, R. Impagliazzo, L. A. Levin, R. Venkatesan, and D. Zuckerman, "Security preserving amplification of hardness," in *FOCS*, vol. I. IEEE, 1990, pp. 318–326.

[4] L. Csirmaz, "An impossibility result on graph secret sharing," *Des. Codes Cryptography*, vol. 53, no. 3, pp. 195–209, 2009.

[5] S. Sumathy and B. U. Kumar, "Secure key exchange and encryption mechanism for group communication in wireless ad hoc networks," *CoRR*, vol. abs/1003.3564, 2010.

[6] V. Ustimenko, "Cryptim: Graphs as tools for symmetric encryption," in *AAECC*, ser. Lecture Notes in Computer Science, S. Boztas and I. Shparlinski, Eds., vol. 2227. Springer, 2001, pp. 278–286.

[7] J. Hughes and A. Tannenbaum, "Length-based attacks for certain group based encryption rewriting systems," in *Sécurité des Communications sur Internet, SECI-02*, 2002.

[8] A. Z. Broder, "Generating random spanning trees," in *FOCS*. IEEE, 1989, pp. 442–447.

[9] D. Aldous, "The random walk construction of uniform spanning trees and uniform labelled trees," *SIAM J. Discrete Math.*, vol. 3, no. 4, pp. 450–465, 1990.

Generación de primos: una perspectiva computacional

R. Durán Díaz
Universidad de Alcalá,
Alcalá de Henares, España
e-mail: raul.duran@uah.es

L. Hernández Encinas
Instituto de Física Aplicada, CSIC,
Madrid, España
e-mail: luis@iec.csic.es

J. Muñoz Masqué
Instituto de Física Aplicada, CSIC,
Madrid, España
e-mail: jaime@iec.csic.es

***Resumen*—En este trabajo presentamos un resumen de los principales métodos utilizados para la generación de primos con particular énfasis en las optimizaciones desarrolladas para dispositivos móviles, que suelen disponer de prestaciones computacionales limitadas. Se presentan, además, los resultados de una implementación en Maple del método que consideramos más optimizado junto con resultados experimentales de su rendimiento.**

I. Introducción

En los últimos años hemos asistido a una expansión sin precedentes en el uso de la criptografía de clave pública, incluso en el ámbito de las relaciones institucionales, como es el caso del DNI electrónico y del pasaporte electrónico. Estos documentos y otros muchos necesitan utilizar números primos (generalmente de gran tamaño) como elementos básicos para generar las claves o para otros estadios del protocolo criptográfico. Resulta, pues, de interés revisar el estado del arte en lo que se refiere a los métodos para generar primos.

Es también un hecho notable la enorme difusión que han conseguido dispositivos móviles o portátiles, pequeños pero dotados de recursos computacionales más o menos amplios, según los casos. Desde teléfonos móviles o PDAs, con alta capacidad de cómputo y de memoria, hasta tarjetas inteligentes, generalmente mucho más modestas en uno y otro recursos, todos ellos son ejemplos de sistemas en donde encontramos con frecuencia capacidades criptográficas. Es necesario diseñar algoritmos de generación de primos apropiados a estos dispositivos, en cuanto a demanda de computación y de almacenamiento.

En este trabajo presentamos un resumen de los principales métodos utilizados para la generación de primos, en dos aplicaciones muy populares, GnuPG y OpenSSL, y en dispositivos móviles con prestaciones computacionales limitadas. Incluimos, además, resultados experimentales.

El trabajo se estructura así. La sección II repasa los algoritmos de primalidad deterministas y probabilísticos, con especial énfasis en el algoritmo de Miller-Rabin. Las secciones III y IV presentan y analizan datos experimentales sobre la generación de primos en dos ámbitos diferenciados. Por un lado se consideran las aplicaciones GnuPG y OpenSSL, sobre plataformas estándar; y por otro, se estudia el algoritmo de Joye-Paillier sobre plataformas móviles. Finalmente, las conclusiones se presentan en la sección V.

II. Algoritmos de detección de primalidad

Es sabido que decidir sobre la primalidad de un número es una tarea mucho más sencilla que factorizarlo, aunque los métodos de comprobación de primalidad implican gran carga computacional. Ordinariamente se basan en comprobar el cumplimiento, por parte del candidato a primo, de ciertas condiciones, cuya verificación implica su primalidad. Con frecuencia se recurre a métodos que exigen comprobaciones menos costosas, pero que, por contra, no dan con toda seguridad una respuesta correcta, es decir, pueden declarar que un número es primo cuando en realidad es compuesto.

Así pues, se llama *test de primalidad* (resp. *test de composición*) a un algoritmo que determina que un candidato es primo (resp. compuesto). Un test de primalidad decide con total seguridad acerca de la primalidad del candidato. Sin embargo, el test de composición sólo es capaz de determinar con toda seguridad que el candidato es compuesto. Por esta razón, a los primeros se les añade el calificativo de *deterministas*, mientras que los segundos se denominan *tests de primalidad probabilísticos*.

II-A. Algoritmos deterministas

El trabajo de Agrawal *et al*. ([1]) estableció que determinar la primalidad de un número admite un algoritmo de primalidad determinista de tiempo polinómico, es decir, el problema de la primalidad es de tipo **P**. Sin embargo, el interés de este algoritmo es más bien teórico, pues su tiempo de ejecución es de $\tilde{O}\left((\log n)^{21/2}\right)$ en el caso más pesimista, aunque puede mejorar a $\tilde{O}\left((\log n)^{6}\right)$ aceptando ciertas conjeturas habituales. La notación $\tilde{O}(x)$ significa $O(x\cdot \mathrm{pol}(\log x))$, donde $\mathrm{pol}(x)$ es un polinomio en x.

Este algoritmo fue rápidamente mejorado por otros autores, entre los que destacan Berrizbeitia ([2]) que baja ese tiempo a $\tilde{O}\left((\log n)^{4}\right)$ para aquellos números primos n para los cuales n^2-1 es divisible por una potencia de 2 cercana a $(\ln n)^2$. Cheng ([3], [4]) extendió ese resultado a una clase más amplia de números primos n, aquellos tales que $n-1$ es divisible por un primo $e \simeq (\log n)^2$. Bernstein ([5]) generaliza el resultado a valores $e \simeq (\log n)^2$ que dividan a n^d-1, con $d \in n^{o(1)}$ (en la práctica, lo interesante es el caso $d=1$). Independientemente, Avanzi y Mihailescu en [6] generalizaron totalmente el valor e.

Todos ellos son algoritmos de tipo aleatorio que proporcionan un tiempo de cómputo esperado de $\tilde{O}\left((\log n)^4\right)$.

Aunque obtener estos resultados ha supuesto un notable esfuerzo, el tiempo de computación de tales algoritmos los descarta de un uso práctico y hay que recurrir a los algoritmos probabilísticos.

II-B. Algoritmos probabilísticos: test de Miller-Rabin

El test de Miller-Rabin es el algoritmo universalmente utilizado para producir los llamados «primos industriales». Este algoritmo cae en la categoría de test de composición, por lo que existe una probabilidad no nula de que el algoritmo califique de «no compuesto» un número que realmente lo es. Sin embargo, el test permite reducir esa probabilidad a un valor tan pequeño como se desee con una eficiencia computacional alta.

El teorema de Fermat es la base de éste y de otros algoritmos probabilísticos de comprobación de primalidad y afirma que si n es primo y a es un entero tal que $\mathrm{mcd}(n,a)=1$, entonces $a^{n-1}\equiv 1 \pmod{n}$. Si se desea comprobar la primalidad de n y se encuentra una base prima con n en que el teorema de Fermat no se verifica, se puede asegurar que n es compuesto. Ahora bien, la recíproca no es cierta: aunque todas las posibles bases verifiquen el teorema, eso no garantiza que el número n sea primo. A pesar de ello, el test de Fermat en sí mismo es extraordinariamente útil por varias razones:

1. La condición del teorema es sólo necesaria, pero el número de excepciones (*pseudoprimos*) es muy bajo.
2. Desde el punto de vista computacional depende casi totalmente de la *exponenciación modular*.

Los números n que satisfacen la congruencia $a^{n-1}\equiv 1 \pmod{n}$ para toda base $a\in[2,n-1]$ tal que $\mathrm{mcd}(a,n)=1$ se conocen como *números de Carmichael*. Por desgracia, existen infinitos de ellos (ver [7]), pero esto no impide seguir usando el teorema de Fermat como se verá ahora.

Si n un número entero impar positivo y a otro entero, se escribe $n-1=2^s q$, con q otro entero impar. En estas condiciones, se dice que n es un *pseudoprimo robusto* en la base a si o bien $a^q\equiv 1 \pmod{n}$ o bien existe un e tal que $0\le e<s$ y $a^{2^e q}\equiv -1 \pmod{n}$.

Observación 1: Si p es un primo impar, es fácil ver que también p es un pseudoprimo robusto en cualquier base a tal que $\mathrm{mcd}(a,p)=1$. Recíprocamente, se puede probar (véase, por ejemplo, [8]) que si p no es primo, existen menos de $p/4$ bases a tales que $1<a<p$ para las cuales p es un pseudoprimo robusto en la base a.

Suponiendo que las probabilidades son independientes, es claro que si un candidato resulta pseudoprimo robusto para t bases aleatorias, la probabilidad de que sea realmente primo será[1] $1-2^{-2t}$. Con esto, Miller ([10]) y Rabin ([11]) desarrollaron el test que lleva su nombre y podemos enunciar como sigue.

[1]Esta probabilidad es demasiado pesimista, al no tener en cuenta la distribución de los primos. Puede verse un análisis más detallado en [9, §§4.48, 4.49], que proporciona valores adecuados para el parámetro de seguridad de acuerdo a la probabilidad deseada. En dicho análisis se demuestra también que el parámetro depende además de la longitud en bits del candidato analizado.

ENTRADA:	$n\in\mathbb{Z}$, parámetro de seguridad $t\in\mathbb{N}$.
SALIDA:	n es compuesto o primo probable con probabilidad $1-2^{-2t}$.

1. [Inicialización]
   Se eligen s, $q\in\mathbb{Z}$ tales que $n-1=2^s q$, q impar.
2. [Lazo]
   `while` $(t>0)$
   { Elegimos un entero $a\in[2,n-1]$ aleatoriamente.
   $b=a^q \pmod{n}$;
   `if` $(b==1$ o bien $b==n-1)$ `goto seguir;`
   `for` $(j\in[1,s-1])$
   { $b=b^2 \pmod{n}$;
   `if` $(b==n-1)$ `goto seguir;`
   `if` $(b==1)$ `return` "n es compuesto";
   }
   `return` "n es compuesto";
   `seguir:`
   $t=t-1$;
   }
   `return` "n es primo con probabilidad $1-2^{-2t}$";

Figura 1. Algoritmo de Miller-Rabin

Algoritmo 2: Dados $n\in\mathbb{Z}$, candidato a comprobar, y $t\in\mathbb{N}$, parámetro de seguridad, este algoritmo determina si n es primo con una probabilidad de acierto de $1-2^{-2t}$. El algoritmo está descrito en la figura 1.

Es importante ahora tratar del tiempo de ejecución de este algoritmo, por estar en el corazón de los generadores de primos utilizados en la práctica. De la observación del algoritmo, en la figura 1 queda claro que el tiempo de ejecución viene dominado por el necesario para llevar a cabo la operación de exponenciación modular. Existen muchos algoritmos para la exponenciación (véanse, por ejemplo, [12, §1.2], [13, Cap. 9], [14, §9.3]) que requieren ordinariamente $O(\log n)$ multiplicaciones en el grupo. Por tanto es fundamental tratar de elegir el método de multiplicar que minimice el tiempo de computación.

Los algoritmos de multiplicación más sencillos requieren $O\left((\log n)^2\right)$ operaciones básicas, entendiendo por tales las que involucran dígitos de un tamaño máximo, digamos B, en bits. Existen otros métodos, sin embargo, que permiten reducir este número.

1. Método de Karatsuba ([15]). Divide los números a multiplicar en dos partes más pequeñas, reduciendo 4 multiplicaciones a 3. Así, el número de operaciones básicas pasa a ser $O\left((\log n)^{\log 3}\right)\simeq O\left((\log n)^{1,585}\right)$.
2. Método de Toom-Cook ([14, §9.5]). Divide los números a multiplicar en k partes más pequeñas (si $k=2$, se convierte en Karatsuba). Cuando, por ejemplo, $k=3$, el número de multiplicaciones pasa de 9 a 5, con lo que el número de operaciones básicas es $O\left((\log n)^{\log 5/\log 3}\right)\simeq O\left((\log n)^{1,465}\right)$.

3. Métodos que involucran transformadas de Fourier, como el de Schönhage y Strassen ([16]) que necesita un número de operaciones de $O\left(n \log n \log^2 n\right)$.

Todos los métodos citados necesitan realizar aparte las reducciones modulares, lo que implica costosas divisiones. Ello se evita usando la reducción de Montgomery ([17], [18]), un ingenioso método de realizar la reducción modular, que evita las divisiones. Este método resulta de mucho interés para los dispositivos móviles por su economía, pero ha sido objeto de intensos ataques de canal lateral como el *timing attack* (véase, por ejemplo, [19]). Se concluye con la siguiente

Proposición 3: El tiempo de ejecución esperado para el algoritmo de Miller-Rabin es $O\left((\log n)^{2+\varepsilon}\right)$.
El valor de ε depende del método de multiplicación empleado.

III. Generadores en GnuPG y OpenSSL

En esta sección estudiamos los generadores utilizados en dos importantes aplicaciones del mundo del software abierto: GnuPG y OpenSSL. Hemos elegido estas dos aplicaciones por acogerse a las licencias de tipo *GNU General Public License*.

GnuPG es la implementación hecha por GNU de OpenPGP, tal como está definido en el RFC4880. Entre las operaciones soportadas está, naturalmente, la generación de claves, que necesita como base una primitiva de generación de números primos. Obviamente, los algoritmos usados para la generación impactan en la calidad de los primos generados y, por tanto, en la seguridad básica del sistema.

OpenSSL es un proyecto para desarrollar en código abierto una implementación de los estándares Secure Sockets Layer (SSL) y Transport Layer Security (TLS), que constituyen la base más usada actualmente para la comunicación segura a través de internet. Protocolos como `https`, para navegación web con autenticación y cifrado, aplicaciones como `ssh` para comunicación serie (terminal serie) cifrada y autenticada, son clientes de tales estándares. También es interesante en este caso estudiar y cualificar la calidad de las primitivas de generación de primos, base para la generación de las claves.

III-A. Generación en GnuPG

GnuPG usa GMP, *GNU Multiple Precision Arithmetic Library* ([20]) como biblioteca de multiprecisión[2], modificada ligeramente en cuanto al modo de almacenar los datos.

La generación de números primos está centralizada en una sola función denominada `gen_prime` y contenida en el fichero `cipher/primegen.c` de la distribución.

El proceso de generación implica dos fases: la generación de un número aleatorio candidato y la comprobación de primalidad mediante tests sucesivamente más fiables y costosos computacionalmente. En todo el proceso, se utiliza una lista estática de números primos hasta e incluyendo 4999, que se prepara antes de comenzar el proceso.

[2]Las bibliotecas de multiprecisión recogen rutinas que permiten utilizar datos de precisión arbitraria (por ejemplo, enteros de tamaño arbitrariamente grande, o números de coma flotante con precisión arbitrariamente pequeña) para lenguajes de programación estándar, como C o C++. En particular, GMP es una biblioteca de código abierto, avalada por una amplísima base instalada.

1. Fase de generación del candidato:
   a) Genera un número aleatorio. Para ello utiliza los recursos del sistema operativo; en particular, para los sistemas tipo Unix/Linux, el dispositivo `urandom`.
   b) Asegura que los bits más y menos significativos sean '1'. Así garantiza un determinado número de bits y que el candidato es impar.
2. Fase de comprobación de primalidad:
   a) Realiza un test de divisiones sobre la tabla de primos bajos.
   b) Realiza un test de Fermat.
   c) Realiza un test de Miller-Rabin, con un parámetro de seguridad fijado en 5 para todos los candidatos.

Si en alguno de los tests de la fase 2 el candidato resulta ser compuesto, se repiten los tests sobre el siguiente impar, hasta probar un total de 10.000. Si todos los candidatos resultan compuestos, se vuelve a la fase 1, se genera un nuevo número aleatorio y se repite todo el proceso.

El algoritmo de exponenciación usado es el clásico de «potencia cuadrada y producto repetidos» ([12, §1.2]) junto con una reducción por simple división con resto. La multiplicación usa el método de Karatsuba si resulta más eficiente.

III-B. Generación en OpenSSL

OpenSSL utiliza una biblioteca propia, incluida en el propio paquete, como biblioteca de multiprecisión. Esta biblioteca se denomina BIGNUM y está escrita en lenguaje C.

En este caso, la generación de primos está centralizada en la función `BN_generate_prime_ex`, contenida en el fichero `crypto/bn/bn_prime.c`. Los clientes de esta función son, por ejemplo, el generador de parámetros para intercambio de claves de Diffie-Hellman, o la generación de claves para RSA.

En este caso, también la generación de primos comporta dos fases: la generación de un número aleatorio y la comprobación posterior mediante un algoritmo de primalidad.

1. Fase de generación del candidato:
   a) Genera un número aleatorio, utilizando funciones de la propia biblioteca, basadas *grosso modo* en la familia MD de funciones resumen (*hash*).
   b) Dentro de la misma fase, comprueba también que el número aleatorio así generado no contenga ningún factor común con alguno de los primos de una tabla de primos estáticamente generada. Dicha tabla está prefijada en el fichero `crypto/bn/bn_prime.h` y alberga los 2048 primeros primos, desde 2 hasta e incluyendo 17863.
2. Fase de comprobación de primalidad:
   a) Obtiene un parámetro de seguridad dependiente del tamaño del candidato, con el fin de garantizar una probabilidad menor de 2^{-80} para cualquier número de bits que tenga el candidato. Específicamente, se elige 3 para los candidatos de 1024 bits, 6 para los de 512 bits y 12 para los de 256 bits (véase [9, Tabla 4.4]).

b) Se realiza un test de Miller-Rabin con el parámetro de seguridad seleccionado en el punto anterior.

Como dato de interés, la biblioteca utiliza la exponenciación de Montgomery, implementada en la función `BN_mod_exp_mont`.

III-C. Resultados experimentales

En esta sección presentamos los resultados experimentales acerca de los tiempos de computación y el número de llamadas al algoritmo de primalidad que han sido necesarios para obtener primos de diversas longitudes utilizando para ello los generadores respectivos de GnuPG y de OpenSSL.

La metodología de trabajo ha sido la misma para las medidas correspondientes a ambas aplicaciones. A continuación, resumimos los pasos dados para las mediciones.

1. Aislar las rutinas que generan los primos en cada aplicación.
2. Crear un programa principal como *envoltorio* capaz de generar un primo de la longitud deseada.
3. Armar una batería de tests que generen 200 números primos de longitudes desde 100 hasta 1000 bits, con saltos de 100 bits, registrando en cada caso los datos de tiempo de ejecución y número de llamadas al algoritmo de primalidad.

Las longitudes se han seleccionado teniendo en cuenta los valores que pueden ser de interés hoy en día. Ello no obstante, los resultados parecen ser fácilmente extrapolables. Para la ejecución de los programas, se ha utilizado una plataforma de tipo Intel Pentium M, a 1,60 gHz, con un tamaño de caché de 2048 kbytes y 1 gbyte de memoria RAM.

Los resultados experimentales se pueden ver en las figuras 2-3. La figura 2 representa el tiempo de ejecución necesario, medido en milisegundos y representado logarítmicamente, frente al número de bits requerido. Hemos representado el promedio de tiempos resultado de la batería de tests ejecutada para cada una de las aplicaciones, de modo que se pueda ver una comparación entre ambas al golpe de vista. Los resultados son muy similares si bien se ve que OpenSSL consigue una

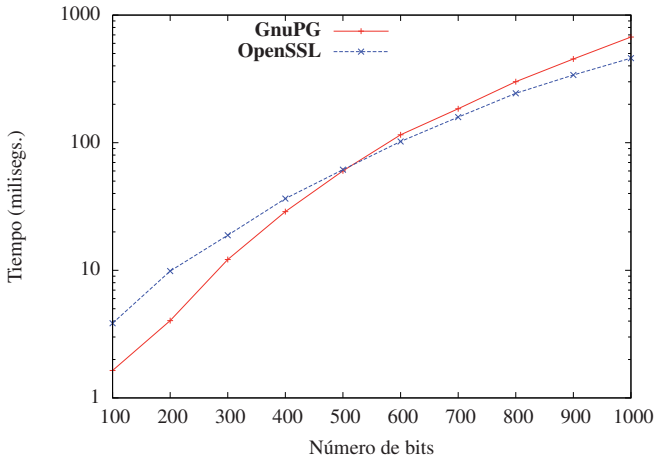


Figura 2. Tiempo de ejecución promediado frente a número de bits

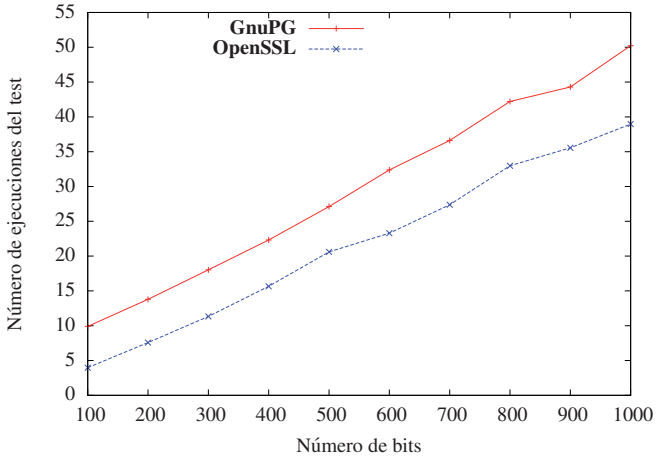


Figura 3. Número promediado de invocaciones al algoritmo de primalidad

pendiente más suave lo que le favorece para el cómputo de números primos más grandes.

La figura 3 representa el promedio de invocaciones al algoritmo de primalidad. Aquí claramente la ventaja es para OpenSSL, que consigue obtener primos con menos invocaciones. Ello se debe a que los candidatos presentados por OpenSSL al algoritmo son de «mejor calidad», es decir, tienen más probabilidades de ser realmente primos. Sin duda esto explica su buen comportamiento en términos de tiempo de ejecución, tal como se acaba de ver en la figura anterior.

IV. Generación de primos en dispositivos móviles

Después de repasar los métodos habituales de generación de primos en plataformas estándar, nos centramos en el problema de hacer lo mismo sobre dispositivos móviles, tales como tarjetas inteligentes, PDAs, etc. Este tipo de dispositivos se caracteriza por su limitación tanto en capacidad de proceso (que es nula en algunos casos) como en capacidad de almacenamiento. Ello hace inviables los métodos habituales y hay que recurrir a métodos especializados que permitan generar primos eficientemente en este tipo de plataformas. Tal es el caso del método de Joye y Paillier ([21], [22]).

La propuesta citada es capaz de producir primos q uniformemente distribuidos en un intervalo prefijado, $[q_{min}, q_{max}]$, donde q_{min} y q_{max} son dos enteros arbitrarios, $q_{min} < q_{max}$. Se supone que el dispositivo está dotado de un generador de números aleatorios y de una función de comprobación de primalidad T. El objetivo es maximizar la velocidad de generación, básicamente reduciendo el número de aplicaciones del test T, que es lo más costoso computacionalmente. Veamos a continuación las distintas fases del algoritmo.

Selección de los parámetros del sistema. Tomamos $0 < \varepsilon \leq 1$ como un parámetro de calidad (por ejemplo, $\varepsilon = 10^{-3}$) y sea φ la función de Euler. Se elige un un conjunto de primos y se calcula $\Pi = \prod_i p_i$, tal que existan enteros t, v, w, que satisfacen

(P1) $1 - \varepsilon < \frac{w\Pi - 1}{q_{max} - q_{min}} \leq 1$;
(P2) $v\Pi + t \geq q_{min}$;

ENTRADA: $v,w,t \in \mathbb{Z}$, y $a \in \mathbb{Z}_m^* \setminus \{1\}$.
SALIDA: primo aleatorio en el intervalo $[q_{min}, q_{max}]$.

1. [Inicialización]
   Calcular $\ell = v\Pi$, $m = w\Pi$.
   Seleccionar aleatoriamente $k \in \mathbb{Z}_m^*$.
   $q = [(k-t) \pmod m] + t + \ell$;
2. [Lazo]
   `while (`$\mathsf{T}(q)$ `== false)`
   { $k = k \cdot a \pmod m$;
   $q = [(k-t) \pmod m] + t + \ell$;
   }
   `return` q;

Figura 4. Generación de primos de Joye-Paillier

(P3) $(v+w)\Pi + t - 1 \leq q_{max}$;
(P4) el cociente $\varphi(\Pi)/\Pi$ es lo más pequeño posible.

Los primos generados están en el intervalo $[v\Pi + t, (v+w)\Pi + t - 1] \subseteq [q_{min}, q_{max}]$. De acuerdo a (P1), cuanto más pequeño sea ε, tanto mejores resultados se obtienen. En (P4), minimizar el cociente $\varphi(\Pi)/\Pi$ garantiza que Π maximiza el número de primos distintos lo más pequeños posible. Dados $(q_{min}, q_{max}, \varepsilon)$, calcular los valores de (Π, v, w, t) que satisfagan las propiedades (P1)-(P4) es experimentalmente sencillo.

Generación de primos. El algoritmo se puede ver en la figura 4. Es de destacar que el primer paso necesita un valor aleatorio $k \in \mathbb{Z}_m^*$, por lo que hace falta un algoritmo que permita la selección computacionalmente simple de unidades. Este algoritmo se presenta en la figura 5. También es importante destacar que tanto a como k están en el grupo de las unidades $\mathbb{Z}_m^*$, por lo que siempre son coprimos con Π. Aquí radica la genialidad del algoritmo: es capaz de generar candidatos a primos que excluyen, *por construcción*, una lista tan grande como se quiera y pueda de factores primos (pequeños). Con esto disminuye proporcionalmente el número de veces que se ha de invocar el test de primalidad antes de obtener un primo.

Generación de unidades. La clave para el buen comportamiento del algoritmo es la posibilidad de generar fácilmente elementos $k \in \mathbb{Z}_m^*$. Para ello, se utilizan los siguientes resultados (demostrados o referenciados en [21]).

ENTRADA: m y $\lambda(m)$, con λ función de Carmichael.
SALIDA: unidad aleatoria $k \in \mathbb{Z}_m^*$.

1. [Inicialización]
   Seleccionar aleatoriamente $k \in [1, m]$.
   $u = (1 - k^{\lambda(m)}) \pmod m$;
2. [Lazo]
   `while (`$u \neq 0$`)`
   { Seleccionar aleatoriamente $r \in [1, m]$.
   $k = k + ru \pmod m$;
   $u = (1 - k^{\lambda(m)}) \pmod m$;
   }
   `return` k;

Figura 5. Generación de unidades de Joye-Paillier

Proposición 4: Sean $m > 1$ y $k \in \mathbb{Z}_m$. Entonces, $k \in \mathbb{Z}_m^*$ si y sólo si $k^{\lambda(m)} \equiv 1 \pmod m$, donde λ es la función de Carmichael.

Recordemos que la función de Carmichael de un número n se define como el entero más pequeño, $\lambda(n)$ tal que $a^{\lambda(n)} \equiv 1 \pmod n$ para todo $a \in \mathbb{Z}_n$ tal que $\mathrm{mcd}(a,n) = 1$. Si $n = p_1^{s_1} \cdots p_t^{s_t}$, entonces se puede calcular recursivamente como $\lambda(n) = \mathrm{mcm}(\lambda(p_1^{s_1}), \ldots, \lambda(p_t^{s_t}))$. A su vez, si $p \geq 3$, o $s \leq 2$, $\lambda(p^s) = \varphi(p^s) = p^{s-1}(p-1)$, y $\lambda(2^s) = 2^{s-2}$.

Proposición 5: Sean $k, r \in \mathbb{Z}_m$, tales que $\mathrm{mcd}(r, k, m) = 1$. Entonces $k + r(1 - k^{\lambda(m)}) \pmod m \in \mathbb{Z}_m^*$.

Está claro que calcular la función de Carmichael de m, $\lambda(m)$, es fácil porque, por construcción, sabemos perfectamente la factorización de m.

IV-A. Resultados experimentales

Para obtener resultados de utilización del algoritmo de Joye-Paillier no hemos podido contar con dispositivos físicos, por lo que hemos recurrido a su implementación en el sistema Maple de álgebra simbólica, que es muy popular y proporciona todas las primitivas necesarias para ello. No adjuntamos el código por falta de espacio.

La metodología ha sido similar a la utilizada para las aplicaciones GnuPG y OpenSSL: el programa implementado en Maple es capaz de generar un primo de la longitud pedida, reportando el tiempo necesario para ello y el número de llamadas al algoritmo de primalidad. Es importante notar que los sistemas de cómputo simbólico se ejecutan a una velocidad relativamente lenta, por lo que los resultados de tiempo de computación son interesantes sólo desde un punto de vista relativo. Con la ayuda de este programa, hemos realizado una batería de tests para generar primos en el intervalo aproximado desde 100 a 1000 bits, en pasos de aproximadamente 100 bits. Para cada longitud se generan 500 primos y se registra el tiempo de computación y el número de llamadas al algoritmo de primalidad necesarios para obtenerlos. Finalmente se calcula el promedio de ambas medidas.

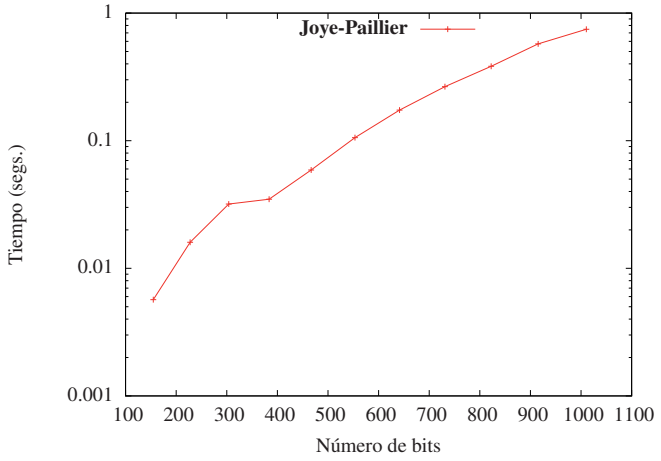


Figura 6. Tiempo de ejecución promediado frente a número de bits

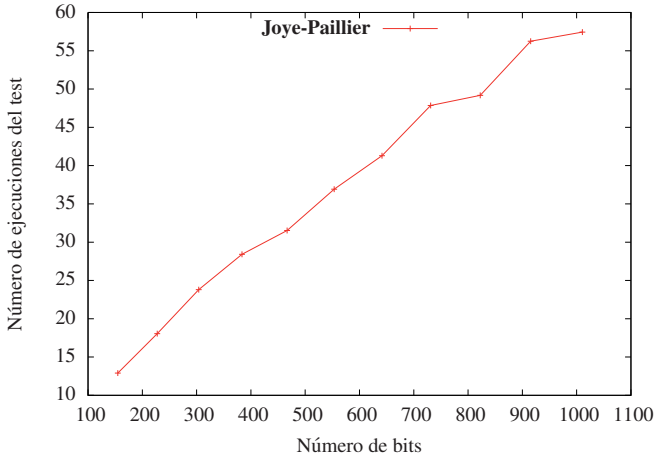


Figura 7. Número promediado de invocaciones al algoritmo de primalidad

Los resultados pueden verse en las figuras 6-7. En la figura 6 se presenta el tiempo medio de computación. Se observa que, en términos relativos, el comportamiento es relativamente similar al que se ha obtenido para las aplicaciones GnuPG y OpenSSL. Como ya se ha indicado, no fue posible realizar estas mediciones sobre dispositivos móviles reales, por lo que nos hemos de limitar a señalar el aspecto de la curva. No obstante pensamos que no es atrevido esperar un comportamiento similar.

En la figura 7 se puede observar que, en general, el número de llamadas al algoritmo de primalidad resulta ser más alto para todo el rango de tamaño en bits si se compara con los resultados obtenidos en la figura 3. Esto resulta en detrimento de este algoritmo que precisamente buscaba minimizar el número de tales llamadas. Ello parece indicar que el buen comportamiento del algoritmo resulta muy dependiente de una correcta elección de los parámetros del sistema.

V. Conclusiones

En este trabajo hemos presentado un resumen de los principales métodos utilizados para la generación de primos en diferentes plataformas. Se ha analizado experimentalmente la eficiencia de los métodos utilizados por dos aplicaciones muy populares, GnuPG y OpenSSL, así como por el algoritmo de Joye-Paillier. Los primeros pueden considerarse de carácter general, mientras que el último está especialmente diseñado para tarjetas criptográficas con limitada capacidad de cómputo.

Hemos diseñado programas específicos que aíslan el proceso de generación de primos para cada una de ellas y presentamos resultados experimentales acerca del tiempo necesario y número de invocaciones al algoritmo de primalidad en cada caso, considerando la generación de primos con distintas longitudes.

Los procesos de generación de primos, en todos los casos, pueden considerarse muy eficientes y los algoritmos empleados garantizan una alta calidad en el proceso de generación de primos. Este resultado es especialmente destacable en el caso del OpenSSL.

Aunque no ha sido posible realizar una implementación sobre dispositivos reales, los resultados experimentales aquí presentados parecen indicar que el algoritmo de Joye-Paillier supone una optimización sobre tarjetas criptográficas, tanto en tiempo como en recursos, en línea con las aplicaciones GnuPG y OpenSSL, destinadas a plataformas estándares.

Agradecimientos

Los autores agradecen a los revisores sus sugerencias para la mejora de este trabajo que ha sido parcialmente financiado por el Ministerio de Ciencia e Innovación mediante el proyecto TEC2009-13964-C04-02, y el Ministerio de Industria, Turismo y Comercio, en colaboración con CDTI y Telefónica I+D mediante el proyecto Segur@ CENIT-2007 2004.

Referencias

[1] M. Agrawal, N. Kayal y N. Saxena, “PRIMES Is in P”, en *Ann. of Math.*, vol. 160, no. 2, pp. 781–793, 2004.
[2] P. Berrizbeitia, “Sharpening ‘Primes is in P’ for a large family of numbers”, en *Math. Comp.*, vol. 74, no. 252, pp. 2043–2059, 2005.
[3] Q. Cheng, “Primality proving via one round in ECPP and one iteration in AKS”, en *Lecture Notes in Comput. Sci.*, vol. 2729, pp. 338–348, 2003.
[4] ——, “Primality proving via one round in ECPP and one iteration in AKS”, en *J. Cryptology*, vol. 20, no. 3, pp. 375–387, 2007.
[5] D. J. Bernstein, “Proving primality in essentially quartic random time”, en *Math. Comp.*, vol. 76, pp. 389–403, 2003.
[6] P. Mihăilescu y R. Avanzi, “Efficient ‘quasi’-deterministic primality test improving AKS”, preprint, http://cacciopoli.mac.rub.de/website/papers/aks-mab.pdf
[7] W. Alford, A. Granville y C. Pomerance, “There are infinitely many Carmichael numbers”, en *Ann. of Math.*, vol. 140, pp. 703–722, 1994.
[8] D. Knuth, *The Art of Computer Programming*. Reading, MA, USA: Addison-Wesley Publishing Co., 1968, 1980, 2nd edition, vol. 2 - Seminumerical Algorithms.
[9] A. Menezes, P. van Oorschot y S. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, Inc., 1997.
[10] G. Miller, “Riemann’s hypothesis and tests for primality”, en *J. Comput. System Sci.*, vol. 13, pp. 300–317, 1976.
[11] M. Rabin, “Probabilistic algorithms for testing primality”, en *J. Number Theory*, vol. 12, pp. 128–138, 1980.
[12] H. Cohen, *A Course in Computational Algebraic Number Theory*. Berlin: Springer, 1993.
[13] R. M. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen y F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, H. Cohen, G. Frey y C. Doche, Eds. Boca Raton, FL, USA: Chapman & Hall/CRC, 2005.
[14] R. Crandall y C. Pomerance, *Prime Numbers. A Computational Perspective*. New York: Springer, 2001.
[15] A. Karatsuba, “The complexity of computations”, en *Proc. Steklov Inst. Math.*, vol. 211, pp. 169–183, January 1995.
[16] A. Schönhage y V. Strassen, “Schnelle Multiplikation großer Zahlen”, en *Computing (Arch. Elektron. Rechnen)*, vol. 7, pp. 281–292, 1971.
[17] P. L. Montgomery, “Modular multiplication without trial division”, en *Math. Comp.*, vol. 44, no. 170, pp. 519–521.
[18] Ç. K. Koç, T. Acar y B. S. Kaliski, Jr., “Analyzing and comparing Montgomery multiplication algorithms”, en *IEEE Micro*, vol. 16, no. 3, pp. 26–33, June 1996.
[19] H. Sato, D. Schepers y T. Takagi, “Exact analysis of Montgomery multiplication”, en *Lecture Notes in Comput. Sci.*, vol. 3348, pp. 290–304, 2004.
[20] T. Granlun, “The GNU multiprecision arithmetic library”, http://gmplib.org, 2010.
[21] M. Joye y P. Paillier, “Fast generation of prime numbers on portable devices: An update”, en *Lecture Notes in Comput. Sci.*, vol. 4249, pp. 160–173, 2006.
[22] M. Joye, P. Paillier y S. Vaudenay, “Efficient generation of prime numbers”, en *Lecture Notes in Comput. Sci.*, vol. 1965, pp. 340–354, 2000.

Un esquema multiusuario de intercambio de clave

Clara Gallardo
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
Email: cgallardo@dccia.ua.es

José Vicent
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
Email: jvicent@dccia.ua.es

Antonio Zamora
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
Email: zamora@dccia.ua.es

***Resumen*—En este artículo se presenta un esquema de intercambio de clave entre n usuarios que requiere $n-1$ envíos entre los participantes. Es una generalización del intercambio de clave para dos usuarios, y está basado en potencias de matrices triangulares superiores por bloques con elementos en $\mathbb{Z}_p$. La seguridad de este esquema queda garantizada puesto que se basa en un problema (bajo ciertas condiciones) intratable como es el problema del logaritmo discreto aplicado al grupo de matrices mencionado anteriormente.**

***Index Terms*—Intercambio de clave, criptografía, matrices por bloques, logaritmo discreto.**

I. Introducción

Un protocolo de intercambio de clave es aquel por el cual dos partes, comúnmente llamadas Alice y Bob, acuerdan una clave secreta para el uso en la subsiguiente comunicación privada. El intercambio de clave es una parte esencial de un sistema de clave pública (véanse [7], [10]). El primer esquema de intercambio de clave publicado, fue introducido por Diffie y Hellman en 1976 (véase [8]), e independientemente por Merkle en 1978 (véase [14]); en él, dos usuarios que quieren intercambiar una clave, acuerdan dos valores de entrada del algoritmo de intercambio de clave: un número primo grande p y un elemento generador g.

La seguridad del intercambio de clave de Diffie-Hellman se basa en la dificultad del problema del logaritmo discreto (DLP) sobre cuerpos finitos (véase [6], [11], [13]). Sea G un grupo cíclico de orden n y α un generador de G. Si $\beta \in G$, el logaritmo discreto de β con respecto a α es el elemento $x \in Z_n$ tal que $\beta = \alpha^x$. Como bien es sabido, el DLP para G consiste en determinar x cuando G, α y β son conocidos.

Para que el DLP sea útil en la construcción de primitivas criptográficas es necesario que el problema sea intratable, en otras palabras, no debería tener una solución en tiempo polinomial. La intratabilidad computacional del DLP depende del grupo cíclico G en el que se esté trabajando [12].

En este trabajo se presenta un esquema de intercambio de clave entre n usuarios, en el que son necesarios $n-1$ envíos de información, de forma que éstos puedan compartir un secreto común y utilizar éste como clave de sesión para el cifrado de la información. El esquema presentado utiliza matrices triangulares superiores por bloques con elementos en $\mathbb{Z}_p$ (véase [4]) y basa su seguridad en el DLP que surge cuando se trabaja con matrices de este grupo. En la sección II y como introducción se recuerda el intercambio de clave para dos usuarios. En la sección III se describe la notación utilizada con la intención de clarificar la descripción del esquema y en la sección IV se presenta, a modo de ejemplo, el intercambio de clave para tres usuarios y su generalización para n usuarios, así como la demostración de que finalmente comparten un secreto. Finalmente, en la sección V, se realiza un análisis de seguridad del algoritmo presentado, llegando a la conclusión de que el esquema propuesto es seguro, puesto que basa su seguridad en el DLP aplicado al grupo especial de matrices triangulares superiores por bloques. Además, es eficiente puesto que se utiliza en todos los cálculos un algoritmo de exponenciación rápida (véase [2]) para el mencionado grupo.

II. Intercambio de clave para dos usuarios

Como se ha comentado, el DLP, elegido con un grupo cíclico G adecuado, proporciona un nivel de seguridad alto a los esquemas criptográficos que lo utilizan. En [3] se presenta un intercambio de clave para dos usuarios que basa su seguridad en el DLP. El conjunto Θ sobre el que se aplica este intercambio es el siguiente:

Sea p un número primo y $r, s \in \mathbb{N}$; se denota por $\mathrm{Mat}_{r\times s}(\mathbb{Z}_p)$ a las matrices de tamaño $r \times s$ con elementos en $\mathbb{Z}_p$, y por $\mathrm{GL}_r(\mathbb{Z}_p)$ y $\mathrm{GL}_s(\mathbb{Z}_p)$ a las matrices invertibles de tamaño $r\times r$ y $s\times s$ respectivamente, también con elementos en $\mathbb{Z}_p$. Consideramos el conjunto Θ de las matrices

$$M = \begin{bmatrix} A & X \\ \boldsymbol{0} & B \end{bmatrix},$$

donde $A \in \mathrm{GL}_r(\mathbb{Z}_p), B \in \mathrm{GL}_s(\mathbb{Z}_p)$ y $X \in \mathrm{Mat}_{r\times s}(\mathbb{Z}_p)$.

Se tiene, para h un entero no negativo, que

$$M^h = \begin{bmatrix} A^h & X^{(h)} \\ \boldsymbol{0} & B^h \end{bmatrix},$$

siendo

$$X^{(h)} = \begin{cases} \boldsymbol{0}, & \text{si } h = 0 \\ \displaystyle\sum_{i=1}^{h} A^{h-i} X B^{i-1}, & \text{si } h \geq 1 \end{cases}.$$

Se describe, para un par de números $x, y \in \mathbb{N}$, la notación:

$$\begin{aligned} A_{xy} &= A_1^x A_2^y, \\ B_{xy} &= B_1^x B_2^y, \\ C_{xy} &= A_1^x X_2^{(y)} + X_1^{(x)} B_2^y. \end{aligned}$$

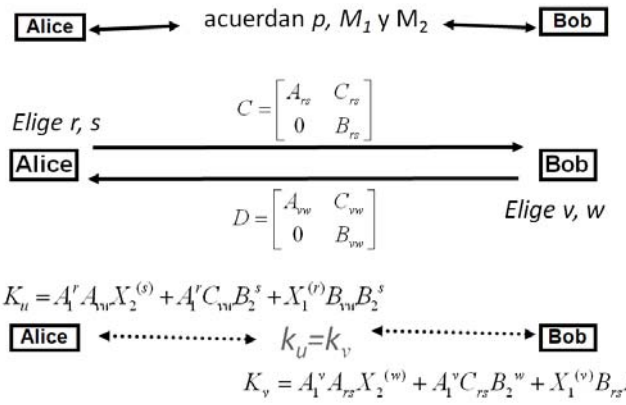


Figura 1. Intercambio de clave para dos usuarios

Si dos usuarios U y V desean intercambiar una clave, deben ejecutar el protocolo siguiente:

1. Acuerdan un número primo p y dos matrices $M_1, M_2 \in \Theta$, con órdenes m_1 y m_2 respectivamente.
2. El usuario U genera dos números aleatorios $r, s \in \mathbb{N}$ tales que $1 \leq r \leq m_1 - 1, 1 \leq s \leq m_2 - 1$, calcula A_{rs}, B_{rs}, C_{rs}, agrupa como bloques en una nueva matriz
$$C = \begin{bmatrix} A_{rs} & C_{rs} \\ \boldsymbol{0} & B_{rs} \end{bmatrix}$$
y envía esta matriz a V.
3. El usuario V genera dos números aleatorios $v, w \in \mathbb{N}$ tales que $1 \leq v \leq m_1 - 1, 1 \leq w \leq m_2 - 1$, calcula A_{vw}, B_{vw}, C_{vw}, agrupa como bloques en una nueva matriz
$$D = \begin{bmatrix} A_{vw} & C_{vw} \\ \boldsymbol{0} & B_{vw} \end{bmatrix}$$
y envía esta matriz a U.
4. El usuario U calcula
$$K_U = A_1^r A_{vw} X_2^{(s)} + A_1^r C_{vw} B_2^s + X_1^{(r)} B_{vw} B_2^s$$
y el usuario V calcula
$$K_V = A_1^v A_{rs} X_2^{(w)} + A_1^v C_{rs} B_2^w + X_1^{(v)} B_{rs} B_2^w,$$
ocurriendo que $K_U = K_V = K$, siendo K el valor de la clave compartida.

Los pasos a seguir para realizar el intercambio de clave con dos participantes quedan resumidos en la figura 1.

III. Notación

A continuación describimos la notación necesaria para realizar la ampliación del intercambio de clave a n participantes.

Sean las matrices
$$M_1 = \begin{bmatrix} A_1 & X_1 \\ \boldsymbol{0} & B_1 \end{bmatrix} \text{ y } M_2 = \begin{bmatrix} A_2 & X_2 \\ \boldsymbol{0} & B_2 \end{bmatrix}$$
elementos del conjunto Θ con órdenes m_1 y m_2 respectivamente.

Se define, para $i \in \{1, \ldots, n\}$,
$$\begin{aligned} A^{1,i} &= A_1^{r_i} A_2^{s_i}, \\ B^{1,i} &= B_1^{r_i} B_2^{s_i}, \\ C^{1,i} &= A_1^{r_i} X_2^{(s_i)} + X_1^{(r_i)} B_2^{s_i}, \end{aligned}$$
y, para $i \in \{1, \ldots, n\}, k \in \{2, \ldots, n\}$, utilizaremos la siguiente notación:
$$\begin{aligned} A^{k,i} &= A_1^{r_i} A^{k-1,i-1} A_2^{s_i}, \\ B^{k,i} &= B_1^{r_i} B^{k-1,i-1} B_2^{s_i}, \\ C^{k,i} &= A_1^{r_i} C^{k-1,i-1} B_2^{s_i} + A_1^{r_i} A^{k-1,i-1} X_2^{(s_i)} + \\ &\quad X_1^{(r_i)} B^{k-1,i-1} B_2^{s_i}, \end{aligned}$$
siendo $r_i, s_i \in \mathbb{N}$ dos números aleatorios generados por el usuario U_i.

En el intercambio de clave que se realizará a continuación, el exponente k deberá interpretarse como el número de envío de información que se realiza, mientras que el exponente i hace referencia al número de usuario que realiza el cálculo, siendo $i - 1$ una manera abreviada de hacer referencia al usuario anterior (en vez de $(i - 2 \bmod n) + 1$).

IV. Intercambio de clave para n usuarios

El objetivo del artículo es ampliar el intercambio de clave presentado en [2] para que participen en él n usuarios. A continuación, utilizaremos la notación expuesta en el apartado anterior para realizar el intercambio de clave para tres participantes y, seguidamente, lo extenderemos a n.

Cabe decir que, cuando se extiende el esquema de intercambio a más de 2 participantes, todos ellos han de saber cuántos usuarios participan en total y en qué orden.

Sean U_1, U_2 y U_3 tres usuarios que desean intercambiar una clave. Para ello ejecutan el protocolo siguiente:

1. Acuerdan p primo y $M_1, M_2 \in \Theta$, con órdenes m_1 y m_2 respectivamente, así cómo el índice $i \in \{1, 2, 3\}$ correspondiente a cada participante.
2. El usuario U_1 genera dos números aleatorios r_1 y $s_1 \in \mathbb{N}$ tales que $1 \leq r_1 \leq m_1 - 1, 1 \leq s_1 \leq m_2 - 1$, calcula $A^{1,1}, B^{1,1}$ y $C^{1,1}$, agrupa estos valores como bloques en una nueva matriz
$$C_1^1 = \begin{bmatrix} A^{1,1} & C^{1,1} \\ \boldsymbol{0} & B^{1,1} \end{bmatrix}$$
y la envía al usuario U_2.
3. El usuario U_2 genera dos números aleatorios r_2 y $s_2 \in \mathbb{N}$ tales que $1 \leq r_2 \leq m_1 - 1, 1 \leq s_2 \leq m_2 - 1$, calcula $A^{1,2}, B^{1,2}$ y $C^{1,2}$, agrupa estos valores como bloques en una nueva matriz
$$C_2^1 = \begin{bmatrix} A^{1,2} & C^{1,2} \\ \boldsymbol{0} & B^{1,2} \end{bmatrix}$$
y la envía al usuario U_3.
4. Por último, el usuario U_3 genera dos números aleatorios r_3 y $s_3 \in \mathbb{N}$ tales que $1 \leq r_3 \leq m_1 - 1, 1 \leq s_3 \leq m_2 - 1$,

calcula $A^{1,3}, B^{1,3}$ y $C^{1,3}$, agrupa estos valores como bloques en una nueva matriz

$$C_3^1 = \begin{bmatrix} A^{1,3} & C^{1,3} \\ \boldsymbol{0} & B^{1,3} \end{bmatrix}$$

y la envía al usuario U_1.

5. Utilizando los bloques de la matriz que ha recibido de U_3, el usuario U_1 calcula $A^{2,1}, B^{2,1}$ y $C^{2,1}$, agrupa estos valores como bloques en una nueva matriz

$$C_1^2 = \begin{bmatrix} A^{2,1} & C^{2,1} \\ \boldsymbol{0} & B^{2,1} \end{bmatrix}$$

y envía esta matriz a U_2.

6. De manera análoga, el usuario U_2 calcula $A^{2,1}, B^{2,2}$ y $C^{2,2}$, agrupa estos valores como bloques en una nueva matriz

$$C_2^2 = \begin{bmatrix} A^{2,2} & C^{2,2} \\ \boldsymbol{0} & B^{2,2} \end{bmatrix}$$

y envía esta matriz a U_3.

7. El usuario U_3 calcula $A^{2,3}, B^{2,3}$ y $C^{2,3}$, agrupa estos valores como bloques en una nueva matriz

$$C_3^2 = \begin{bmatrix} A^{2,3} & C^{2,3} \\ \boldsymbol{0} & B^{2,3} \end{bmatrix}$$

y envía esta matriz a U_1.

8. Los tres participantes U_1, U_2 y U_3 calculan

$$\begin{aligned} K_{U_1} &= A_1^{r_1} C^{2,3} B_2^{s_1} + A_1^{r_1} A^{2,3} X_2^{(s_1)} + \\ &\quad X_1^{(r_1)} B^{2,3} B_2^{s_1}, \end{aligned}$$

$$\begin{aligned} K_{U_2} &= A_1^{r_2} C^{2,1} B_2^{s_2} + A_1^{r_2} A^{2,1} X_2^{(s_2)} + \\ &\quad X_1^{(r_2)} B^{2,1} B_2^{s_2} \end{aligned}$$

y

$$\begin{aligned} K_{U_3} &= A_1^{r_3} C^{2,2} B_2^{s_3} + A_1^{r_3} A^{2,2} X_2^{(s_3)} + \\ &\quad X_1^{(r_3)} B^{2,2} B_2^{s_3}, \end{aligned}$$

ocurriendo que $K_{U_1} = K_{U_2} = K_{U_3} = K$, que es la clave compartida, como se demostrará más adelante.

A continuación se generaliza el intercambio para el caso en el que participen n usuarios.

Sea $n \in \mathbb{N}$, y sean $U_i, i \in \{1 \dots n\}$, interlocutores que desean intercambiar una clave. Para ello ejecutan el protocolo siguiente:

1. Acuerdan un número primo p y $M_1, M_2 \in \Theta$, con órdenes m_1 y m_2 respectivamente, así cómo el índice $i \in \{1 \dots n\}$ correspondiente a cada participante.
2. Cada participante U_i genera dos números aleatorios r_i y $s_i \in \mathbb{N}$ tales que $1 \leq r_i \leq m_1 - 1, 1 \leq s_i \leq m_2 - 1$.
3. Repetir, desde $k = 1$ hasta $k = n - 1$
   3.1 Cada usuario U_i calcula $A^{k,i}, B^{k,i}$ y $C^{k,i}$ y agrupa estos valores como bloques en una nueva matriz

$$C_i^k = \begin{bmatrix} A^{k,i} & C^{k,i} \\ \boldsymbol{0} & B^{k,i} \end{bmatrix}. \qquad (1)$$

   3.2 El participante U_i envía esta matriz C_i^k al usuario siguiente, esto es, el usuario U_i envía C_i^k al usuario U_j, siendo $j = (i \bmod n) + 1$.
4. Utilizando la última matriz recibida, cada participante U_i calcula

$$\begin{aligned} K_{U_i} &= A_1^{r_i} C^{n-1,i-1} B_2^{s_i} + A_1^{r_i} A^{n-1,i-1} X_2^{(s_i)} + \\ &\quad X_1^{(r_i)} B^{n-1,i-1} B_2^{s_i} = K, \end{aligned}$$

que es la clave compartida.

El teorema siguiente muestra que la clave es común a todos los participantes.

Teorema 1: $K_{U_i} = K_{U_{i-1}} = K, i = 2 \dots n$.

Proof: Para $i = 2 \dots n$, sea $P_i = M_1^{r_i} M_2^{s_i}$, con

$$M_1^{r_i} = \begin{bmatrix} A_1^{r_i} & X_1^{(r_i)} \\ \boldsymbol{0} & B_1^{r_i} \end{bmatrix} \text{ y } M_2^{s_i} = \begin{bmatrix} A_2^{s_i} & X_2^{(s_i)} \\ \boldsymbol{0} & B_2^{s_i} \end{bmatrix},$$

y sea

$$M_{U_i} = M_1^{r_i} P_{i-1} M_2^{s_i} = \begin{bmatrix} A_{U_i} & K_{U_i} \\ \boldsymbol{0} & B_{U_i} \end{bmatrix}.$$

Con esto se tiene que

$$\begin{aligned} M_{U_i} &= M_1^{r_i} P_{i-1} M_2^{s_i} \\ &= M_1^{r_i} M_1^{r_{i-1}} M_2^{s_{i-1}} M_2^{s_i} \\ &= M_1^{r_{i-1}} M_1^{r_i} M_2^{s_i} M_2^{s_{i-1}} \\ &= M_1^{r_{i-1}} P_{u_i} M_2^{s_{i-1}} \\ &= M_{U_{i-1}} \end{aligned}$$

y, consecuentemente, $K_{U_i} = K_{U_{i-1}}, i = 2 \dots n$. ■

El teorema que se muestra a continuación explicita la forma de la clave compartida.

Teorema 2: La clave que se comparte con el intercambio descrito tiene la forma

$$\begin{aligned} K_{U_i} &= A_1^{r_i} C^{n-1,i-1} B_2^{s_i} + A_1^{r_i} A^{n-1,i-1} X_2^{(s_i)} + \\ &\quad X_1^{(r_i)} B^{n-1,i-1} B_2^{s_i}. \end{aligned}$$

Proof: Se demuestra por inducción sobre k, siendo k el número de envío.

Para $k = 2$. El usuario U_{j-1} calcula $A^{1,j-1}$, $B^{1,j-1}$, $C^{1,j-1}$, agrupa los valores como en (1) (con $k = 1$) y envía esta matriz a U_j.

U_j calcula

$$M_1^{r_j} C_{j-1}^1 M_2^{s_j} =$$

$$\begin{bmatrix} A_1^{r_j} & X_1^{(r_j)} \\ \boldsymbol{0} & B_1^{r_j} \end{bmatrix} \begin{bmatrix} A^{1,j-1} & C^{1,j-1} \\ \boldsymbol{0} & B^{1,j-1} \end{bmatrix} \begin{bmatrix} A_2^{s_j} & X_2^{(s_j)} \\ \boldsymbol{0} & B_2^{s_j} \end{bmatrix} =$$

$$\begin{bmatrix} A_1^{r_j} A^{1,j-1} & A_1^{r_j} C^{1,j-1} + X_1^{(r_j)} B^{1,j-1} \\ \boldsymbol{0} & B_1^{r_j} B^{1,j-1} \end{bmatrix} \begin{bmatrix} A_2^{s_j} & X_2^{(s_j)} \\ \boldsymbol{0} & B_2^{s_j} \end{bmatrix} =$$

$$\begin{bmatrix} A^{2,j} & C^{2,j} \\ \boldsymbol{0} & B^{2,j} \end{bmatrix} = C_j^2,$$

donde

$$C^{2,j} = A_1^{r_j} A^{1,j-1} X_2^{(s_j)} + (A_1^{r_j} C^{1,j-1} + X_1^{(r_j)} B^{1,j-1}) B_2^{s_j}$$

y, por tanto, para $k = 2$ el bloque superior derecho tiene la forma indicada.

Supongamos ahora que el bloque superior derecho de las matrices que se mandan en el envío $n-1$ tienen esa forma, y veamos que la que se forma a partir de ellas (cuyo bloque superior derecho es la clave compartida) también la tiene.

Sea entonces

$$C_{j-1}^{n-1} = \begin{bmatrix} A^{n-1,j-1} & C^{n-1,j-1} \\ \boldsymbol{0} & B^{n-1,j-1} \end{bmatrix}$$

la matriz calculada por el usuario U_{j-1} para realizar la comunicación $n-1$. Éste la envía al usuario siguiente, U_j, que calcula

$$M_1^{r_j} C_{j-1}^{n-1} M_2^{s_j} =$$

$$\begin{bmatrix} A_1^{r_j} & X_1^{(r_j)} \\ \boldsymbol{0} & B_1^{r_j} \end{bmatrix} \begin{bmatrix} A^{n-1,j-1} & C^{n-1,j-1} \\ \boldsymbol{0} & B^{n-1,j-1} \end{bmatrix} \begin{bmatrix} A_2^{s_j} & X_2^{(s_j)} \\ \boldsymbol{0} & B_2^{s_j} \end{bmatrix} =$$

$$\begin{bmatrix} A_1^{r_j} A^{n-1,j-1} A_2^{s_j} & K_{U_j} \\ \boldsymbol{0} & B_1^{r_j} B^{n-1,j-1} B_2^{s_j} \end{bmatrix},$$

siendo

$$K_{U_j} = A_1^{r_j} A^{n-1,j-1} X_2^{(s_j)} + (A_1^{r_j} C^{n-1,j-1} + X_1^{(r_j)} B^{n-1,j-1}) B_2^{s_j}$$

la clave compartida, que tiene la forma indicada.

■

V. Análisis de Seguridad

Para evitar ataques por fuerza bruta, el orden de M_1 y M_2 tiene que escogerse suficientemente grande (del orden de 1024 bits).

El algoritmo de Menezes y Wu [13] es una técnica común para analizar la seguridad de esquemas de clave pública en los que intervienen potencias de matrices. Básicamente, este algoritmo establece la posibilidad de reducir el problema del logaritmo discreto a una serie de logaritmos discretos sobre un cuerpo finito de pequeño tamaño $F_{q^{m_i}}$, donde m_i es el grado del polinomio irreducible, que es un factor del polinomio característico de la matriz A. La entrada de este algoritmo es una matriz cuadrada, como $A, B \in GL_n(\mathbb{Z}_p)$ con $B = A^l$, y la salida es el exponente entero l. Así pues, este algoritmo puede ser efectivo contra esquemas que hacen públicas potencias de matrices cuadradas. Sin embargo en el esquema presentado sólo se hacen públicas las matrices C_i^k, lo que hace inviable este tipo de ataques.

Climent, Gorla y Rosenthal [5] proponen una técnica de ataque basada en el teorema de Cayley-Hamilton, que es susceptible de ser usada en esquemas donde aparecen matrices triangulares superiores por bloques. Este ataque está basado en la existencia de un único polinomio característico, lo cual invalida su utilización en el esquema presentado, ya que se usan dos matrices M_1 y M_2 con polinomios caracteristicos diferentes.

González, Pérez y Taborda [9] realizan un análisis para el intercambio de clave de dos usuarios habiendo realizado una reducción del problema a una extensión del cuerpo base, llegando a la conclusión de que el esquema matricial no presenta ventajas con respecto a la computación en dicho cuerpo. Esto último condiciona la elección del primo p de manera que el DLP sea lo suficientemente duro.

VI. Conclusiones

En este artículo se ha descrito un esquema de intercambio de clave entre n usuarios en el que son necesarios $n-1$ envíos de información. El esquema presentado se basa en un intercambio de clave para dos usuarios que utiliza un grupo especial de matrices triangulares superiores por bloques (con unas excelentes propiedades criptográficas). El intercambio de clave propuesto, basa su seguridad en el DLP aplicado al citado grupo de matrices. Además es un protocolo eficiente, puesto que se utiliza un algoritmo de exponenciación rápida para matrices por bloques.

Referencias

[1] M. Abdalla, M. Bellare, P. Rogaway, "The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES", CT-RSA 2001, LNCS, vol. 2020, Springer, Berlin, pp. 143-158, 2001.

[2] R. Alvarez, F. Ferrández, J. Vicent, A. Zamora, "Applying Quick Exponentiation for Block Upper Triangular Matrices", Applied Mathematics and Computation, Vol. 183, 729-737, 2006.

[3] R. Alvarez, L. Tortosa, J. Vicent, A. Zamora, "Analysis and Design of a Secure Key Exchange Scheme", Information Sciences, Vol. 179, 2014-2021, 2009.

[4] R. Alvarez, L. Tortosa, J. Vicent, A. Zamora, "A Non-Abelian Group Based on Block Upper Triangular Matrices with Cryptographic Applications", Lecture Notes in Computer Science, Vol. 5527, 117-126., 2009.

[5] J.J. Climent, E. Gorla, J. Rosenthal, "Cryptanalysis of the CFVZ Cryptosystem", Advances in Mathematics of Communications (AMC), Vol. 1, 1-11, 2006.

[6] D. Coppersmith, A. Odlyzko, R. Schroeppel, "Discrete Logarithms in GF(p)", Algorithmica, 1-15, 1986.

[7] R. Cramer, V. Shoup, "Design and Analysis of Practical Public-key Encryption Schemes Secure Against Adaptive Chosen Ciphertext Attack", SIAM J. Comput. 33(1), 167-226, 2003.

[8] W. Diffie, M. E. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, IT-22, 6, 664-654, 1976.

[9] M.I. González, A. Pérez, P. Taborda, "Cryptanalysis of a Key Exchange Scheme Based on Block Matrices", Cryptology ePrint Archive (IACR), 2010.

[10] G. Hanaoka, K. Kurosawa, "Efficient Chosen Ciphertext Secure Public Key Encryption under the Computational Diffie-Hellman Assumption", ASIACRYPT 2008, pp. 308-325, 2008.

[11] K. McCurley, "The Discret Logarithm Problem", Cryptology and Computational Number Theory, Proceedings of Symposia in Applied Mathematics, Vol. 42, 49-74, 1990.

[12] A. Menezes, P.C. van Oorschot, S.A. Vanstone, "Handbook of Applied Cryptography", The CRC Press Series on Discrete Mathematics and Its Applications, Boca Raton, 1997.

[13] A. Menezes, Y.H. Wu, "The Discrete Logarithm Problem in GL(n, q)", Ars Combinatoria, Vol. 47, 22-32, 1997.

[14] R. Merkle, "Secure Communications over Insecure Channels", Communications of the ACM vol. 21, 4, 294-499, 1978.

Identity-based non-interactive key distribution with forward security

Rainer Steinwandt
Department of Mathematical Sciences
Florida Atlantic University
Boca Raton, FL 33426, U.S.A.
Email: rsteinwa@fau.edu

Adriana Suárez Corona
Departamento de Matemáticas
Universidad de Oviedo
33007 Oviedo, Spain
Email: adriana@orion.ciencias.uniovi.es

***Abstract*—Identity-based non-interactive key distribution (ID-NIKD) is a cryptographic primitive that enables two users to establish a common secret key without exchanging messages. All users of the system have access to public system parameters and a private key, obtained through the help of a trusted key generation center.**

In this contribution, we discuss how to capture an intuitive form of *forward security* for ID-NIKD schemes in a security model. Building on results of Sakai et al. as well as of Paterson and Srinivasan, we discuss how the proposed notion of forward security can be achieved in the random oracle model, using a Bilinear Diffie-Hellman assumption in combination with a forward-secure pseudorandom bit generator.

I. Introduction

Identity-based non-interactive key distribution (ID-NIKD) has already been discussed by a number of authors – including work by Blom [3], Matsumoto and Imai [5], Tsujii et al. [10], Maurer and Yacobi [6], [7], and Dupont and Enge [4]. Starting point for our work is the security model for ID-NIKD proposed by Paterson and Srinivasan [8], specifically the following comment in the latter paper: *we note that no non-interactive key distribution scheme can meet the notion of* forward security *that is enjoyed by many interactive key distribution protocols.* Subsequently we explore the question of achieving forward security, by passing from an "ordinary" ID-NIKD scheme to a primitive to which we refer as *key evolving* ID-NIKD, which is in line with the *key evolution paradigm* as discussed for signatures by Bellare and Miner [1].

After recalling the relevant technical definitions in the next section, in Section III we provide the definition of a key evolving ID-NIKD along with a security model, capturing forward security. Thereafter, we discuss how a combination of a forward-secure pseudorandom bit generator with an ID-NIKD scheme by Sakai et al. [9] can be used to achieve forward security in the sense we defined it.

II. Preliminaries

On the cryptographic side, a main technical tool we need is a *forward-secure pseudorandom bit generator* and, building on Sakai et al.'s proposal, we also make use of pairings.

A. Stateful pseudorandom bit generators

For the relevant facts on pseudorandom bit generators, we follow mainly the exposition of Bellare and Yee [2].

Definition 1 (stateful pseudorandom bit generator): A *stateful generator* $\mathsf{GEN} = (\mathsf{GEN.key}, \mathsf{GEN.next}, b, m)$ is a tuple of polynomial time algorithms as follows:

- GEN.key is probabilistic, and on input the security parameter outputs the initial state (seed).
- GEN.next is deterministic, and given the current state St_i outputs a pair (Out_{i+1}, St_{i+1}), where Out_{i+1} is a b-bit string and St_{i+1} is the next state.
- b is the size of the output blocks.
- m is the maximum number of output blocks the generator may be used to produce.

In order to achieve forward security of a generator as specified in the above definition, we require that every time the generator outputs a pair (Out_i, St_i), the previous state St_{i-1} is erased, so that an adversary breaking into the system is only able to learn the current state.

Definition 2 (forward secure pseudorandom bit generator): A stateful generator GEN is *forward-secure* if the advantage of any probabilistic polynomial time adversary $\mathcal{A}$ attacking GEN as described in Figure 1 is negligible. For this, the advantage of $\mathcal{A}$ is defined as $\mathrm{Adv}^{\mathrm{fsprg}}_{\mathsf{GEN}}(\mathcal{A}) =$

$$|\Pr[\mathrm{Exp}^{\mathrm{fsprg}-1}_{\mathsf{GEN}}(\mathcal{A}) = 1] - \Pr[\mathrm{Exp}^{\mathrm{fsprg}-0}_{\mathsf{GEN}}(\mathcal{A}) = 1]|$$

In other words, we require the outputs generated in the past by the stateful generator to be computationally indistinguishable from (true) random bits. The adversary can decide, depending on the output blocks it has seen, when to "break in", i. e., to try to distinguish the generator's output from a true random source. During the find stage ($\mathcal{A}(\mathsf{find}, Out, h)$), every time $\mathcal{A}$ receives an output block, it outputs a pair (h, d), where h is the updated history and d indicates if $\mathcal{A}$ prefers to remain in the find stage or rather wants to guess if the outputs have been generated by the generator (Experiment 1) or are random strings (Experiment 0).

In [2] Bellare and Yee show how to build a forward-secure stateful generator from a secure standard pseudorandom bit generator and from number-theoretic assumptions.

Experiment $\mathrm{Exp}^{\mathrm{fsprg}-1}_{\mathsf{GEN}}(\mathcal{A})$
$St_0 \xleftarrow{\$} \mathsf{GEN.key}$
$i \leftarrow 0;\ h \leftarrow \epsilon$
`Repeat`
  $i \leftarrow i+1$
  $(Out_i, St_i) \leftarrow \mathsf{GEN.next}(St_{i-1})$
  $(d, h) \xleftarrow{\$} A(\mathsf{find}, Out_i, h)$
`Until` $(d = \mathsf{guess})$ **or** $i = m$
$g \xleftarrow{\$} A(\mathsf{guess}, St_i, h)$
`Return` g

Experiment $\mathrm{Exp}^{\mathrm{fsprg}-0}_{\mathsf{GEN}}(A)$
$St_0 \xleftarrow{\$} \mathsf{GEN.key}$
$i \leftarrow 0;\ h \leftarrow \epsilon$
`Repeat`
  $i \leftarrow i+1$
  $(Out_i, St_i) \leftarrow \mathsf{GEN.next}(St_{i-1})$
  $Out_i \xleftarrow{\$} \{0,1\}^b$
  $(d, h) \xleftarrow{\$} \mathcal{A}(\mathsf{find}, Out_i, h)$
`Until` $(d = \mathsf{guess})$ **or** $i = m$
$g \xleftarrow{\$} \mathcal{A}(\mathsf{guess}, St_i, h)$
`Return` g

Fig. 1. Attacking forward security of a stateful generator

.

B. Pairings

For the relevant terminology on pairings, we follow mainly Paterson and Srinivasan [8].

Definition 3 (pairing): Let G be an additive group and G_T a multiplicative group, both of primer order q. Denote by P a generator of G. A *pairing* is a map $e : G \times G \longrightarrow G_T$ with the following properties:

- *Bilinearity:* For all $Q, R \in G$ and for all $a, b \in \{1, \ldots, q-1\}$ we have $e(aQ, bR) = e(Q, R)^{ab}$.
- *Non-degeneracy:* $e(P, P) \neq 1$.
- *Computability*: There is a probabilistic polynomial time algorithm to compute $e(Q, R)$ for all $Q, R \in G$

Note that the map e is symmetric, since $e(aP, bP) = e(P, P)^{ab} = e(bP, aP)$.

To express the hardness we want to use in the protocol discussed below, we need a generator algorithm $\mathbf{PairingGen}$ that on input the security parameter 1^k outputs, keeping the above notation, a tuple (G, G_T, e, q, P) with $q > 2^k$.

Definition 4 (BDH*):* Given a generator algorithm $\mathbf{PairingGen}$ as just described, for a probabilistic polynomial time algorithm $\mathcal{A}$ we define the advange of $\mathcal{A}$ in solving the *Bilinear Diffie-Hellman problem (*BDH*)* as

$$\mathsf{Adv}^{\mathrm{bdh}}_{\mathcal{A}}(k) = \Pr(\mathcal{A}(aP, bP, cP) = e(P, P)^{abc}),$$

where $a, b, c \xleftarrow{\$} \{0, \ldots, q-1\}$ are chosen uniformly at random.

If $\mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{A}}$ is negligible for all probabilistic polynomial time algorithms $\mathcal{A}$, we say that the BDH *assumption* holds for (G, G_T, e) (or, more precisely, for **PairingGen**).

III. Key-evolving non-interactive key distribution

Building on the definition of an "ordinary" identity-based non-interactive key distribution scheme used in [8], we suggest the following notion of *key evolving* $\mathsf{ID\text{-}NIKD}$, allowing a non-interactive key distribution scheme to have algorithms that compute a secret key of a time period taking as input the secret key of the previous one, both for the central authority and the users. Every time the time period changes, the previous secret key used as input is erased.

Definition 5 (key evolving $\mathsf{ID\text{-}NIKD}$*):* A *key-evolving non-interactive key distribution scheme* is a quintuple of polynomial time algorithms as follows:

- **Setup** is probabilistic and run by a central authority. Given the security parameter 1^k and a total number of time periods $N \in \mathbb{N}$, **Setup** generates an initial master secret key mk^0 of the central authority, along with the public parameters $params$. The public parameters include the description of the private key space $\mathcal{SK}$ and the shared key space $\mathcal{SHK}$.
- **MasterKeyUpdate** is deterministic and run by a central authority. Given the public parameters $params$ and the master secret key mk^{i-1} of the previous period it generates the master secret key mk^i of the current period and deletes mk^{i-1}.
- **KeyExtract** is probabilistic and run by a central authority. Given the current master secret key mk^i, the public parameters $params$ and an identifier $ID \in \{0,1\}^*$, it generates a secret key S^i_{ID} from the private key space $\mathcal{SK}$.
- **KeyUpdate** is deterministic and run by a user. Given the public parameters $params$ and the secret key S^{i-1}_{ID} of the previous period, it generates the secret key S^i_{ID} of the current period and deletes S^{i-1}_{ID}.
- **SharedKey**: is deterministic and run by a user. Given the public parameters $params$, the secret key $S^i_{ID_A}$ of the current period for an identity ID_A and an identifier $ID_B \in \{0,1\}^*$, **SharedKey** generates a key K^i_{AB} from the space of shared keys $\mathcal{SHK}$ specified in $params$.

We require that for any identities ID_A, ID_B and corresponding private keys S_{ID_A}, S_{ID_B} and for any period i, **SharedKey** satisfies the constraint

$$\begin{aligned}&\mathbf{SharedKey}(params, S^i_{ID_A}, ID_B) \quad = \\ &\mathbf{Sharedkey}(params, S^i_{ID_B}, ID_A).\end{aligned}$$

This ensures that the users corresponding to identities ID_A and ID_B can compute the same key without any interaction in every time period i.

- **Setup** The challenger $\mathcal{C}$ runs the **Setup** and hands the resulting public parameters $params$ to the adversary $\mathcal{A}$. The initial master secret key mk^0 remains private, i. e., $\mathcal{C}$ does not forward the value mk^0 to $\mathcal{A}$.
- **Phase 0 (Find)** The adversary is allowed to make the following queries:
  - Extract(ID, i): The challenger $\mathcal{C}$ responds by running the algorithm **MasterKeyUpdate** of the key-evolving ID-NIKD scheme with input mk_0 until the master key of time period i, mk^i is obtained. Then $\mathcal{C}$ runs the algorithm **Extract** with input $(params, mk^i, ID)$ and hands the resulting secret key $S^i_{ID}$to $\mathcal{A}$.
  - Reveal(ID_A, ID_B, i): The challenger $\mathcal{C}$ responds by running algorithm **MasterKeyUpdate** of the key-evolving ID-NIKD scheme with input mk_0 until the master key of time period i, mk^i is obtained. Then $\mathcal{C}$ runs the algorithm **Extract** with input $(params, mk^i, ID_A)$ to get a secret key $S^i_{ID_A}$ and then **SharedKey** with input $(params, S^i_{ID_A}, ID_B)$ to get a shared key K^i_{AB}. Finally, $\mathcal{C}$ hands the value K^i_{AB} to $\mathcal{A}$.
  - Test(ID_A, ID_B, i): The challenger $\mathcal{C}$ computes K^i_{AB} in the same way as for answering a Reveal query. Moreover, $\mathcal{C}$ chooses a random bit $b \xleftarrow{\$} \{0,1\}$. If $b = 0$, $\mathcal{C}$ hands K^i_{AB} to $\mathcal{A}$. Otherwise $(b = 1)$, the challenger $\mathcal{C}$ hands a uniformly at random chosen element from $\mathcal{SHK}$ to $\mathcal{A}$.

  The queries of $\mathcal{A}$ must satisfy the following restrictions:
  - Only one Test query can be made.
  - All inputs i of Reveal and Extract queries must satisfy $i \geq i_{test}$, i. e., only future keys may be compromised.
  - No Reveal query can be made on input $(ID_A, ID_B, i_{\text{test}})$, nor $(ID_B, ID_A, i_{\text{test}})$, if the Test query's input is $(ID_B, ID_A, i\text{test})$.
  - No Extract query can be made on input (ID_A, i_{test}), nor (ID_B, i_{test}), if the Test query's input is $(ID_B, ID_A, i_{\text{test}})$.
- **Phase 1 (Guess)** The adversary outputs a value $b' \in \{0,1\}$ and wins if and only if $b = b'$.

Fig. 2. Attacking forward security of a key-evolving NIKD scheme.

Now, to capture the security of a key-evolving ID-NIKD, we build on the security model used by Paterson and Srinivasan in [8], the main difference being the key evolution component in our model: In a scheme that is secure in the sense of Definition 6 below, an adversary cannot distinguish between a past shared key established between two users and a random element from $\mathcal{SHK}$—*even having the current secret keys of these users*. Forward security is therewith achieved in the sense that shared keys will not be compromised, even if the private keys of users are compromised in the future.

Definition 6 (forward-secure key-evolving ID-NIKD*):* A key-evolving non-interactive key distribution scheme is *forward secure* if the advantage of any probabilistic polynomial time adversary $\mathcal{A}$ in the game described in Figure 2 is negligible for all $N \in \mathbb{N}$. Here the *advantage* of an adversary $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{\text{fs}}_{\mathcal{A}}(k) = \left| \Pr[b = b'] - \frac{1}{2} \right|$$

It is not hard to see that a scheme that is forward-secure in the sense of Definition 6 meets security in the sense of Paterson and Srinivasan's *indistinguishability of shared key (*IND-SK*)* [8], as our model includes adversaries that play the game in the initial state (no key updates are made).

Remark 1: Notice that the restrictions of the model in [8] hold, as only one Test query can be made, no Reveal query can be made on the same input (ID_A, ID_B) as the Test query's, nor on input (ID_B, ID_A), and no Extract query can be made on input ID_A, nor ID_B, as the Test query has the initial (current) state as input.

IV. Realizing forward-secure key-evolving identity-based non-interactive key distribution

In this section we describe a key-evolving ID-NIKD scheme in the random oracle model. Our construction is based on a proposal of Sakai et al. [9], incorporating key update algorithms as required in Definition 5. To so, in addition to a generator algorithm $\mathbf{PairingGen}$ we need a forward-secure pseudorandom bit generator as described in Section II. For the parameter m in Definition 1, we assume $m \geq N$, i. e., that m is not smaller than the total number of time periods N and for the parameter b we assume $2^b \geq q$. Finally, we make use of random oracles

$$\begin{array}{lccc} H_1: & \{0,1\}^* & \longrightarrow & G \\ H_2: & G_T & \longrightarrow & \mathcal{SHK} = \{0,1\}^n \\ H_3: & \{0,1\}^b & \longrightarrow & \{0,\ldots,q-1\}. \end{array}$$

The five comprising algorithms are as follows:

- **Setup**: On input the security parameter 1^k and the total number of time periods N, **PairingGen** is used to obtain (G, G_T, e, q, P). In addition, **Setup** chooses an element $s_0 \xleftarrow{\$} \{0,\ldots,q-1\}$ uniformly at random, specifies a pseudorandom generator GEN $=$

$(\mathsf{GEN.key}, \mathsf{GEN.next}, b, m)$ and matching random oracles H_1, H_2, H_3. Let $r_0 \in \{0,1\}^*$ be the seed generated by $\mathsf{Gen.key}$ and $\mathcal{SHK} = \{0,1\}^n$, where $n \geq k$.

Setup outputs the public parameters $params = (G, G_T, e, P, P_0 = s_0 \cdot P, H_1, H_2, H_3, \mathsf{GEN.next}, m, b, n)$ and the initial master secret key $mk^0 = (s_0, r_0)$

- **MasterKeyUpdate**: On input the previous master key $mk^{i-1} = (s_{i-1}, r_{i-1})$, **MasterKeyUpdate** computes $\mathsf{GEN.next}(r_{i-1}) = (Out_i, St_i)$ and outputs the current master secret key

$$mk^i = (H_3(Out_i)s_{i-1}, St_i) = (s_i, r_i).$$

- **KeyExtract**: On input the current master secret key $mk^i = (s_i, r_i)$ and an identifier ID this algorithm outputs the secret key

$$S^i_{ID} = (s_i H_1(ID), r_i).$$

- **KeyUpdate**: On input a secret key $S^{i-1}_{ID} = (s_{i-1}H_1(ID), r_{i-1})$, this algorithm computes $\mathsf{GEN.next}(r_{i-1}) = (Out_i, St_i)$ and outputs the new secret key

$$S^i_{ID} = (H_3(Out_i)s_{i-1}H_1(ID), St_i) = (s_i H_1(ID), St_i).$$

- **SharedKey**: On input a current private key $S^i_{ID_A}$ and an identifier $ID_B \in \{0,1\}^*$, where $ID_B \neq ID_A$, this algorithm outputs

$$K^i_{AB} = H_2(e(s_i H_1(ID_A), H_1(ID_B))).$$

Making use of the bilinearity and symmetry of the map e, it is not difficult to verify that the above collection of algorithms indeed consitutes a correct key-evolving ID-NIKD scheme in the sense that users A and B will obtain identical keys when executing **SharedKey** with $(S^i_{ID_A}, ID_B)$ and $(S^i_{ID_B}, ID_A)$ respectively.

Moreover, the above scheme turns out to be secure in the sense of Definition 6, provided that the underlying pseudorandom bit generator is forward-secure:

Proposition 1: If GEN is a forward secure pseudorandom bit generator and the BDH assumption holds for the generator algorithm **PairingGen**, the above key-evolving ID-NIKD is forward-secure in the sense of Definition 6.

A proof of this result will be given in the full version of this paper, and here we restrict to saying that the proof splits into two different cases: For an adversary not quering Extract or Reveal for time periods $i > i_{\mathrm{Test}}$, the situation is similar as in [8], and a successful adversary can be used to attack the BDH problem for (G, G_T, e). For the case where an adversary queries Extract or Reveal for time periods $i > i_{\mathsf{Test}}$, a successful attacker can be used to mount an attack against the forward security of GEN.

V. Conclusion

In this contribution we introduced the notion of *key-evolving identity-based non-interactive key distribution* which allows to capture an intuitive form of forward security for ID-NIKD schemes—in a spirit similar to forward-secure signature schemes. As a concrete example of a scheme that fulfills the proposed security requirement, we gave a construction in the random oracle model, using pairings, based on a proposal of Sakai et al. and a forward-secure pseudorandom bit generator.

Acknowledgments

We thank Ignacio Cascudo, Madeline González Muñiz and Kashi Neupane for interesting discussions. The second author acknowledges support of FICYT (project IB-08-147) and Spanish MEC (project MEC-07-MTM2007-67884-C04-01 and FPU grant AP2007-03141, cofinanced by the European Social Fund).

References

[1] Mihir Bellare and Sara K. Miner. A Forward-Secure Digital Signature Scheme. In M. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 431–448. Springer-Verlag, 1999.

[2] Mihir Bellare and Bennet Yee. Forward-Security in Private-Key Cryptography. In M. Joye, editor, *Topics in Cryptology – CT-RSA 2003, The Cryptographers' Track at the RSA Conference 2003*, volume 2612 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, 2003.

[3] Rolf Blom. Non-public key distribution. In D. Chaum, R.L. Rivest, and A.T. Sherman, editors, *advances in Cryptology – CRYPTO '82*, pages 231–236. Plenum, 1983.

[4] Régis Dupont and Andreas Enge. Provably secure non-interactive key distribution based on pairings. *Discrete Applied Mathematics*, 154(2):270–276, 2006.

[5] Tsutomu Matsumoto and Hideki Imai. On the KEY PREDISTRIBUTION SYSTEM: A Practical Solution to the Key Distribution Problem. In C. Pomerance, editor, *Advances in Cryptology – CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 185–193. Springer-Verlag, 1988.

[6] Ueli M. Maurer and Yacov Yacobi. Non-interactive Public-Key Cryptography. In D.W. Davies, editor, *Advances in Cryptology – EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 498–507. Springer-Verlag, 1991.

[7] Ueli M. Maurer and Yacov Yacobi. A Remark on a Non-interactive Public-Key Distribution System. In R.A. Rueppel, editor, *Advances in Cryptology – EUROCRYPT '92*, volume 658 of *Lecture Notes in Computer Science*, pages 458–460. Springer-Verlag, 1993.

[8] Kenneth G. Paterson and Sriramkrishnan Srinivasan. On the relations between non-interactive key distribution, identity-based encryption and trapdoor discrete log groups. *Designs, Codes and Cryptography*, 52:219–241, 2009.

[9] Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairing. In *The 2000 Symposium on Cryptography and Information Security*, pages 26–28, 2000.

[10] Shigeo Tsujii, Kaoru Kurosawa, and Toshiya Itoh. New noninteractive identity-based key distribution system. *Electronic Letters*, 24(22):1356–1357, October 1988.

PODER (PrOponer, DEterminar y Refinar) un Criptoanálisis sobre el Generador Auto-Shrinking

M. E. Pazo Robles
I.T.B.A.
Instituto Tecnológico de Buenos Aires
Av. E. Madero 399, Buenos Aires, Argentina
E-mail: eugepazorobles@gmail.com

A. Fúster Sabater
Instituto de Física Aplicada
C.S.I.C.
Serrano 144, 28006 Madrid, España
E-mail: amparo@iec.csic.es

***Abstract*—En este trabajo se presenta un ataque criptoanalítico sobre el generador auto-shrinking, un generador de secuencia cifrante bien conocido por sus buenas propiedades criptográficas. Se propone un refinamiento de la técnica criptoanalítica de Guess-and-Determine, con supuestos definidos y elaborados a lo largo del propio proceso. Se presentan resultados numéricos que mejoran otros criptoanálisis planteados sobre dicho generador. Concretamente, se han logrado complejidades del orden de $O(2^{0.2L})$ para la cantidad de secuencia interceptada, $O(L)$ para memoria consumida y $O(2^{0.5L})$ para tiempo de ejecución (siendo L la longitud del registro del generador). Se propone asimismo un hardware específico para un criptoanálisis de corte práctico.**

I. Introducción

Los sistemas de cifrado en flujo son los más rápidos dentro de los métodos de cifrado actuales, de ahí que se utilicen en numerosas aplicaciones prácticas como, por ejemplo, los algoritmos A5 (en su doble versión A5/1 y A5/2) que se emplean en telefonía móvil GSM [5], el algoritmo E0 usado en especificaciones de Bluetooth [1] o el algoritmo RC4 utilizado en Microsoft Word y Excel [11]. Un sistema de cifrado en flujo está compuesto por un algoritmo o generador de secuencia cifrante (conocido públicamente) y una clave de cifrado (conocida únicamente por los dos comunicantes). Para cifrar, el emisor realiza una operación OR-exclusiva bit a bit entre la secuencia cifrante y el texto claro (mensaje original), dando lugar al texto cifrado que es el que se va a enviar por el canal de información. Para descifrar, el receptor genera la misma secuencia cifrante que suma bit a bit con el texto cifrado recibido y recupera así el texto claro original. Muchos algoritmos de cifrado en flujo están basados en Registros de Desplazamiento con Realimentación Lineal (LFSRs) [4] cuyas secuencias de salida, las *PN*-secuencias, se combinan entre sí mediante algún procedimiento o función no lineal para producir una secuencia pseudoaleatoria de aplicación criptográfica ([2], [3], [8]).

Los procedimientos de cifrado de flujo se utilizan para dar seguridad criptográfica a sistemas de comunicaciones con requerimientos de velocidad y sincronismo. Uno de estos ejemplos de cifrador en flujo es el generador *auto-shrinking*. La Unión Europea, a través del Proyecto Stork [12], propuso a la comunidad científica internacional romper este generador mediante alguna técnica criptoanalítica que mejorase el ataque tipo TMTO (Time Memory Trade Off). En este trabajo, se presenta un técnica de criptoanálisis efectiva aplicada sobre dicho generador, para distintas longitudes L de su registro de desplazamiento con $L \leq 120$. Concretamente, se ha desarrollado un método basado en trabajos anteriores realizados sobre el generador auto-shrinking, en particular [14], logrando mejorar en varios órdenes de magnitud lo alcanzado en dicho trabajo. Esta mejora permite asegurar que el generador puede romperse en tiempo real. La complejidad de memoria necesaria es muy pequeña, concretamente de $O(L^2)$, mientras que la complejidad de secuencia interceptada (datos) tiene un orden menor que $O(2^{0.2L})$. Por otra parte, la complejidad en tiempo que hemos logrado es menor que $O(2^{0.5L})$. De hecho, haber logrado disminuir este valor, en comparación con el de otros autores, es lo que hace que nuestro criptoanálisis pueda realizarse en tiempo real con un hardware dedicado.

II. El Generador Auto-Shrinking

El generador auto-shrinking fue diseñado por Meier y Staffelbach [7] para su uso en aplicaciones criptográficas. Este generador es de fácil implementación y consiste en un único LFSR de L etapas y polinomio de realimentación primitivo. Este registro genera una única secuencia pseudoaleatoria, $\{s_n\}$, la cual se decima de forma irregular dando origen a la secuencia *auto-shrunken*, $\{z_n\}$, o secuencia de salida del generador. La regla de decimación es extremadamente simple; se consideran pares (s_{2i}, s_{2i+1}) $(i = 0, 1, 2, \ldots)$ de bits consecutivos de $\{s_n\}$ tales que:

1) Si $s_{2i} = 1$, entonces $z_j = s_{2i+1}$.
2) Si $s_{2i} = 0$, entonces s_{2i+1} se rechaza.

Es decir, si el primer bit del par considerado es un 1, entonces el segundo bit se inserta en la secuencia de salida. Por el contrario, si el primer bit del par considerado es un 0, entonces el segundo bit se rechaza. De esta manera, se van eliminando determinados bits de la secuencia $\{s_n\}$ y los que quedan constituyen la secuencia $\{z_n\}$ o secuencia auto-shrunken. La clave de este generador es el estado inicial del LFSR. De acuerdo con [7], el periodo, la complejidad lineal y las propiedades estadísticas de la secuencia $\{z_n\}$ son muy adecuados para su aplicación en criptografía.

La secuencia $\{s_n\}$, generada por el LFSR, puede considerarse formada por dos secuencias diferentes $\{c_n\}$ y $\{b_n\}$

TABLE I
$h(x)$ PARA DIFERENTES POLINOMIOS $P_c(x)$ DE GRADO L

L	$P_c(x)$	$h(x)$
36	$x^{36}+x^{25}+1$	$6, 13, 24$
40	$x^{40}+x^{38}+x^{22}+x^{20}+1$	$11, 29, 30$
52	$x^{52}+x^{49}+1$	$2, 25, 28$
100	$x^{100}+x^{37}+1$	$19, 32, 69$
278	$x^{278}+x^{273}+1$	$3, 137, 142$
455	$x^{455}+x^{341}+x^{230}+x^{116}+1$	$3, 4, 62, 118,$ $171, 176, 228,$ $229, 287, 343, 401$

que responden a los bits de subíndices pares e impares, respectivamente, de la secuencia $\{s_n\}$. Es decir:

$$c_i = s_{2i} \quad \forall\, i \geq 0 \tag{1}$$
$$b_i = s_{2i+1} \quad \forall\, i \geq 0. \tag{2}$$

A su vez, estas dos secuencias corresponden a la misma *PN*-secuencia $\{s_n\}$ generada por el LFSR pero desplazadas entre sí una distancia de 2^{L-1} bits. La existencia de dicho desplazamiento permite expresar una secuencia en función de la otra [14]. Así vemos que:

$$b_i = \sum_{j=0}^{L-1} h_j\, c_{i+j}, \tag{3}$$

donde los h_j son los coeficientes de $h(x)$ un polinomio de grado $L-1$ y coeficientes binarios definido por:

$$h(x) \equiv x^{2^{L-1}} mod\ P_c(x), \tag{4}$$

siendo $P_c(x)$ el polinomio característico del LFSR.

De esta forma expresamos los bits impares de la secuencia $\{s_n\}$ que son los b_i en función de los bits pares que son los c_{i+j}. Nótese que calcular el polinomio $h(x)$ para valores de L elevados no es trivial. De hecho, en este trabajo se ha desarrollado un programa ad-hoc basado en propiedades de la aritmética modular. En la Tabla I se presentan los resultados obtenidos para distintos polinomios característicos de grado L. La representación del polinomio $h(x)$, por ejemplo en el caso $6, 13, 24$, corresponde a $h(x) = x^6 + x^{13} + x^{24}$.

III. ATAQUE PROPUESTO

Este método criptoanalítico se encuadra dentro de las técnicas que poseen una componente estadística y una algebraica. Como vimos anteriormente dentro de la secuencia $\{s_n\}$ podemos distinguir otras 2 secuencias $\{c_n\}$ y $\{b_n\}$. Romper el generador significa hallar la condición inicial para las secuencias $\{c_n\}$ y $\{b_n\}$ o un punto de las mismas a partir del cual es posible continuar generando la secuencia $\{z_n\}$.

A. Idea general

La idea general del ataque se concreta en los siguientes puntos: a partir del conocimiento de N bits de la secuencia $\{z_n\}$ (bits *interceptados*) y de la suposición de l bits de la secuencia $\{c_n\}$, en concreto $(c_0, c_1, \ldots, c_{l-1})$, se plantea un sistema de ecuaciones lineales vía la ecuación (3) para determinar:

1) Los restantes bits de $\{c_n\}$, notados $(c_l, c_{l+1}, \ldots, c_{L-1})$.
2) Los correspondientes de la secuencia $\{b_n\}$, notados $(b_0, b_1, \ldots, b_{L-1})$.

Una vez conocidos los L primeros bits de cada secuencia se genera una cierta cantidad de secuencia auto-shrunken y se compara con los bits de secuencia interceptada. Si el resultado coincide, entonces el generador está roto. Si el resultado no coincide, nos desplazamos un bit sobre la secuencia interceptada y repetimos el proceso. Si no encontramos solución tras los sucesivos desplazamientos sobre los bits interceptados, entonces tomamos una nueva suposición de l bits de la secuencia $\{c_n\}$ y repetimos el proceso. Veamos un ejemplo sencillo.

Ejemplo 1: Sea un LFSR con polinomio característico $P_c(x) = x^5 + x^3 + 1$. Las secuencias generadas para una condición inicial del registro son:
$\{c_n\} = \{1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, \ldots\}$
$\{b_n\} = \{0, \underline{0}, 1, 1, 0, \underline{1}, 0, 0, \underline{1}, \underline{0}, \underline{0}, 0, 0, 1, 0, 1, \underline{0}, \underline{1}, 1, 1, \ldots\}$
$\{z_n\} = \{0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, \ldots\}$.

Para $l = 3$, conocemos $N = 5$ bits interceptados que comienzan en z_0 y consideramos, por ejemplo, la suposición $(c_0, c_1, c_2) = (1, 0, 1)$. Con estos datos planteamos el sistema de ecuaciones y si es posible lo resolvemos. En efecto, los bits que nos restan son (c_3, c_4) y $(b_0, b_1, b_2, b_3, b_4)$. Al ser $c_0 = 1$, sabemos que $b_0 = z_0 = 0$. Como $c_1 = 0$, entones b_1 se desconoce. Hay que tener presente que sólo los bits $c_i = 1$ introducen ecuaciones en el sistema. En este ejemplo estamos considerando una suposiciń con $q = 1$ cero. A continuación, como $c_2 = 1$, entonces $b_2 = z_1 = 1$. A partir de la relación de recurrencia lineal del LFSR, el polinomio $h(x) = x^2 + x^3$ y la ecuación (3), planteamos el siguiente sistema de ecuaciones.

$$\begin{aligned} b_0 &= c_2 + c_3 \\ b_1 &= c_3 + c_4 \\ b_2 &= c_0 + c_3 + c_4 \\ b_3 &= c_0 + c_1 + c_3 + c_4 \\ b_4 &= c_0 + c_1 + c_2 + c_3 + c_4. \end{aligned}$$

Dado que a partir de la suposición de partida, se han despejado (b_0, b_2), podemos resolver el sistema de ecuaciones anterior obteniendo los valores de (c_3, c_4) y (b_1, b_3, b_4). Una vez conocidas las condiciones iniciales: $(c_0, c_1, c_2, c_3, c_4) = (1, 0, 1, 1, 1)$ y $(b_0, b_1, b_2, b_3, b_4) = (0, 0, 1, 1, 0)$, se genera una porción de secuencia cifrante $\{z_{resul}\} = (0, 1, 1, 0, 0, \ldots)$, se comparan los bits generados con los bits interceptados y si coinciden, como ocurre en este ejemplo, entonces podemos decir que el sistema está roto.

B. Algoritmo criptoanalítico

Introducimos primeramente alguna notación adicional:

$\{C_k\}_0^{l-1}$ es la k-ésima suposición para los primeros l bits de la secuencia $\{c_n\}$ con $(1 \leq k \leq Q_{real})$, siendo Q_{real} el número total de suposiciones que van a analizarse.

Can representa la cantidad óptima de bits que se consideran para una comparación válida entre bits interceptados y bits generados.

A continuación vemos el esquema del algoritmo programado:

Input: Polinomio $P_c(x)$ de grado L, l, segmento de N bits interceptados, puntero z al primer bit interceptado z_0, polinomio $h(x)$ y bloque de suposiciones $\{C_k\}_0^{l-1}$ con $(1 \leq k \leq Q_{real})$.

Inicializar: $l \cong L/2$, $Can = 2,5 * L$.

For $k = 1 : |Q_{real}|$ para cada suposición $\{C_k\}_0^{l-1}$:

While puntero $z < N$

Paso 1: Plantear el sistema de ecuaciones lineales.

Paso 2: Resolver el sistema ecuaciones dejando a lo sumo un número n (acotado) de bits sin resolver.

Paso 3: Determinar los valores de $(c_0, c_1, \ldots, c_{L-1})$ y $(b_0, b_1, \ldots, b_{L-1})$. Generar Can bits de secuencia $\{z_n\}$.

Paso 4: Comparar bits interceptados con bits generados. Si son iguales *break*.

Paso 5: $z = z + 1$.

end

$k = k + 1$ se toma una nueva suposición, $z = 0$

end

Output: $(c_0, c_1, \ldots, c_{L-1})$ y $(b_0, b_1, \ldots, b_{L-1})$ con los que es posible continuar generando secuencia auto-shrunken.

Con respecto a este ataque, si bien la idea es sencilla y por tanto efectiva, hay ciertas observaciones que merecen ser tenidas en cuenta.

Observación 1: Cuanto mayor sea el número de ceros en cada suposición, menor la cantidad de ecuaciones que podrán considerarse en el sistema de ecuaciones y, por tanto, mayor la cantidad de bits no resueltos.

Observación 2: No todas las suposiciones son aptas para la resolución del sistema, por lo que hay que seleccionarlas adecuadamente. Esta selección de suposiciones es lo que hace de nuestro criptoanálisis un ataque más efectivo que el que utiliza la técnica de Guess and Determine [14].

Observación 3: Definir el bloque de suposiciones óptimas para el generador en cuestión implica no sólo que sea el bloque con la menor cantidad de suposiciones posibles, sino también que sea aquél que logra resolver la mayor cantidad de bits de $(c_0, c_1, \ldots, c_{L-1})$. De este modo se logra un equilibrio entre complejidad en tiempo y complejidad de datos interceptados.

C. Cálculo del Bloque de Suposiciones

Una forma de abordar la selección de suposiciones es tal y como se realiza en [14] y [9], es decir considerando una cantidad muy elevada de ellas. En este trabajo por el contrario el número de suposiciones es mucho menor puesto que elegimos sólo las suposiciones óptimas, esto es aquéllas que sean capaces de resolver la mayor cantidad de bits en $(c_0, c_1, \ldots, c_{L-1})$.

Por otro lado, para una *PN*-secuencia binaria, vemos que la distancia entre bloques no superpuestos de $\{C_k\}_0^{l-1}$ con q ceros dependerá del grado L del polinomio característico del generador. Esta distancia es la que nos dirá cuántos bits de secuencia interceptada son necesarios para romper el criptosistema con una alta probabilidad de éxito P_{exito}.

Es importante notar que, en ocasiones, parte de los bits $(c_0, c_1, \ldots, c_{L-1})$ no van a resolverse, ya que el sistema de ecuaciones que se plantea no permite la resolución de todas las incógnitas.

La resolución óptima del sistema de ecuaciones es sensible a las ecuaciones que lo conforman. Si en la suposión $\{C_k\}_0^{l-1}$ hay un cero en c_i, entonces no se conoce el correspondiente valor de b_i y esa ecuación no forma parte del sistema. La idea de la optimización es descartar suposiciones que posean un cero que impida contar con alguna ecuación considerada importante, dado que a través de ella es posible resolver una mayor cantidad de bits.

Concluímos que no todas las ecuaciones inciden por igual en la resolución del sistema. La manera de limpiar el total de posibles suposiciones $\{C_k\}_0^{l-1}$ con un número q de ceros en sus bits es seleccionar de entre ellas, sólo las que posean un 1 en la ecuación que interesa.

IV. Resultados Numéricos

El tiempo de procesamiento se computa considerando el número de suposiciones que se utilizan en el ataque y la cantidad de bits que componen el bloque de secuencia interceptada. Se tendría así el número total de intentos para romper el criptosistema, es decir el orden de complejidad en tiempo. Hay que tener presente que sólo son necesarios L bits de secuencia interceptada para plantear el sistema de ecuaciones y romper el generador. El problema es que no sería realista considerar que el sistema rompe el generador precisamente en el primer bit de la secuencia interceptada disponible. Por ello se toman N bits interceptados y nos desplazamos dentro del bloque hasta encontrar la solución correcta. Esto es, el estado del LFSR que nos permite seguir generando secuencia cifrante $\{z_n\}$.

A. Comparación con otros Criptoanálisis

A partir de los resultados numéricos obtenidos, se han construido las siguientes Tablas comparativas. En la Tabla II aparecen representados los resultados de este trabajo para distintos valores de L y de los parámetros: Q_{real} cantidad de suposiciones, N cantidad en bits de secuencia interceptada, Bnr número de bits sin resolver, C_T complejidad en tiempo, C_M complejidad en memoria, C_D complejidad de datos interceptados y P_{exito} porcentaje de éxitos obtenidos. Para cada valor de L se han considerado del orden de 10 polinomios característicos distintos, promediándose los resultados.

En la Tabla III se comparan para un L fijo los parámetros utilizados por varios autores en cuanto a longitud l de la suposición, número de suposiciones Q_{real}, longitud de secuencia interceptada N y cantidad de ceros q en cada suposición. Se aprecia que en [14] la cantidad de ceros en cada suposición es

TABLE II
L VERSUS DIFERENTES ÓRDENES DE MAGNITUD

L	Q_{real}	N	Bnr	$C_T < O(2^{0.5*L})$	$C_M < O(2^L)$	$C_D < O(2^{0.25*L})$	P_{exito}
36	1736	500	2	$O(2^{11} * 2^9)$	L^2	2^9	87%
40	2736	950	3	$O(2^{11} * 2^{10})$	L^2	2^{10}	100%
52	12376	5500	4	$O(2^{14} * 2^{12})$	L^2	2^{12}	85%
100	462411533	2^{21}	$12-13$	$O(2^{28} * 2^{21})$	L^2	2^{21}	97%
120	2^{34}	2^{25}	11	$O(2^{34} * 2^{25})$	L^2	$2^{25} = O(2^{0.2*L})$	97%

mayor y, por tanto, la cantidad de bits no resueltos aumentará lo que traerá consigo una mayor complejidad en tiempo.

En la Tabla IV se comparan las complejidades C_T, C_M y C_D para los diferentes autores. La diferencia principal del presente trabajo con el expuesto en [14] es que Zhang *et al.* tratan de resolver todas las ecuaciones posibles, con el inconveniente de que quedarían muchos bits sin determinar. En cambio en nuestro trabajo, sólo resolvemos aquellas ecuaciones que minimizan la cantidad de bits no resueltos. De este modo podemos calcular que la cota máxima para complejidad en tiempo es $O(2^{0.6L})$, mientras que para [14] esta cota asciende a $O(2^L)$. En este punto hay que señalar también que hay polinomios $P_c(x)$ que tienen mejores prestaciones a la hora de resolver el sistema de ecuaciones, dejando menor cantidad de coeficientes c_i sin resolver.

V. IMPLEMENTACIÓN HARDWARE

Esta sección no pretende ser más que un esbozo de una implementación hardware del critpoanálisis desarrollado en las secciones anteriores sobre el generador auto-shrinking. Se pretende dejar planteada una arquitectura de cara a un desarrollo futuro. Los programas se han escrito en Matlab logrando todos los objetivos perseguidos. Sin embargo, cabe destacar que para un criptoanálisis con tiempos de procesamiento bajos, además del método propuesto y de los programas realizados, hay que llevar a cabo una programación óptima con los medios que hoy por hoy la tecnología permite y nuestros conocimientos también. El propósito fundamental de implementar el critptoanálisis mediante programación en paralelo es minimizar el tiempo que se tarda en recuperar la condición inicial del registro. Desarrollar así este criptoanálisis, require una programación adecuada y un medio físico para implementarla. La programación adecuada implica una orientación al bit, eliminando tiempos espúreos y memoria innecesaria. Todo programa corriendo en entornos poco controlables, como puede ser un sistema operativo, ya no sería óptimo. La solución, que logra aunar todas estas condiciones, es lo que se denomina Lógica Programable o más comúnmente FPGA, Fast Programmable Gate Array. Esta lógica programable está orientada al bit, no posee un sistema operativo y todos sus componentes trabajan a velocidades próximas al reloj al que la lógica está conectada. Siendo conservadores, estos relojes están en el rango de 40-100 Mhz y vienen instalados en distintas placas de desarrollo conjuntamente con las FPGAs [13]. La programación en paralelo es propia de este tipo de lógica y por tanto sería nuestra elección a la hora de un desarrollo con propósitos criptoanalíticos. Los componentes lógicos programables de una FPGA pueden tener la funcionalidad de puertas lógicas básicas como AND, OR, XOR, NOT o incluso funciones combinacionales más complejas, tales como decodificadores o funciones matemáticas simples. Consisten en un arreglo de bloques lógicos programables (CLB: Configurable Logic Block) y canales de ruteo también programables. La implementación de un LFSR mediante FPGAs es muy sencilla y puede llevarse a cabo utilizando configuraciones ya disponibles en los chips FPGA. Se aprecia que implementar un generador Shrinking o un generador auto-shrinking en este tipo de hardware no requiere de gran esfuerzo, ya que el mismo entorno de la FPGA está diseñado para admitir este tipo de generadores.

Por otra parte, la frecuencia de salida de los bits de secuencia cifrante sería sumamente elevada y estaría dada por el ocilador del reloj que se conecta a los flip-flops. Para tener una noción de velocidades, este reloj estaría oscilando a unos 40-100 Mhz en placas de desarrollo de FPGAs de muy bajo coste, siendo más veloces en placas menos económicas.

Hasta el momento hemos hablado del criptoanálisis de este generador, pero como ya dijimos los tiempos computacionales están en un orden de $O(2^{0.5L})$ y este valor se mantiene para todo L. Para establecer sólo magnitudes de tiempos de computación se realizó una comparación sobre procesos muy simples, corriendo en diferentes lenguajes de programación como son el Matlab y el Labview y se extrapoló el tiempo que tardaría el proceso en correr sobre una placa FPGA (NI-RIO-EVAL-101National Semiconductors) [10]. Para realizar esta extrapolación se compararon muchos procesos corriendo en Labview y luego programados en la lógica FPGA. La relación entre tiempos dio como resultado que la placa FPGA tarda entre 100 y 500 veces menos que la programación en Labview. Debemos decir que Labview es un herramienta de análisis orientada a señales digitales, manejo de bits, puertas lógicas, medición de tiempos precisos, estadística y otras funciones. Al estar orientada al bit, mejora la prestación que Matlab puede otorgar para la función que precisamos. Es un lenguaje que se puede traducir a la lógica programable con relativa facilidad y, por tanto, disminuye el tiempo de la puesta en marcha de cualquier programa para que corra en una placa de desarrollo FPGA. Veamos ahora la comparación en cuanto a tiempos de procesamiento para un proceso particular como sería: generar el conjunto de suposiciones que hemos desarrollado para

TABLE III
COMPARACIÓN DE PARÁMETROS POR AUTOR PARA $L = 40$

$L = 40$	Q_{real}	N	q
Mihaljevic, $l = 20$	$2^{L-l} = 1048576$	10^6	5
Pazo Robles, $l = 20$	2736	$700 - 800$	5
Zhang, $l = 25$	2^{22}	$O(2^8)$	9-10

TABLE IV
COMPLEJIDADES POR AUTOR

Autor	C_T	C_M	C_D
Mihaljevic	$O(2^{0.5*L})$	$O(L)$	$O(2^{0.5*L})$
Pazo Robles	$O(2^{0.5*L}) - O(2^{0.6*L})$	$O(L^2)$	$< O(2^{0.25*L})$
Zhang	$O(2^{0.7*L}) - O(2^L)$	$O(L^2)$	$O(2^{0.2*L})$

nuestro criptoanalálisis. Comparamos Matlab, Labview y la extrapolación para la placa FPGA. Para generar una suposición en cada entorno programado vemos lo siguiente:

- *Tiempos en Matlab:* El proceso de cada suposición tarda 15 mseg.
- *Tiempos en Labview:* El proceso de cada suposición tarda 0,5 mseg.
- *Tiempos previstos en la placa FPGA:* El proceso de cada suposición tarda 0,001 mseg, es decir 1 μseg.

Pasar de trabajar en Matlab a Labview significó una disminución de 30 veces en el tiempo de generar suposiciones. Por otro lado pasar de Labview a FPGAs creemos que significará al menos una disminución de entre 100 y 500 veces el tiempo previsto en Labview. Esta relación sale de la experiencia de otros ingenieros que ya han hecho este tránsito. Por supuesto, dependerá de la capacidad de programar en paralelo el proceso concreto que se quiere correr, pero ronda esas magnitudes. Vemos en la Tabla V la comparación entre tiempos de programas para generar suposiciones para distintos valores de L. Todos los programas pueden realizarse bajo una programación en paralelo, lo que supondría bajar los tiempos del modo que se ha planteado. Creemos que dará resultados muy positivos y que el generador auto-shrinking a estas alturas está roto en tiempo real para un $L < 100$.

TABLE V
TIEMPOS PARA GENERAR SUPOSICIONES SEGÚN L Y ENTORNO DE PROGRAMACIÓN

L	Ns	$Matlab$	$Labview$	$FPGA$
36	1736	26 seg	$0,86$ seg	$1,736$ seg
40	2736	41 seg	$1,4$ seg	$2,736$ seg
52	12376	186 seg	$6,2$ seg	$12,375$ mseg
100	462411533	80 días	$2,5$ días	$7 - 8$ min

VI. CONCLUSIONES

En este trabajo se ha logrado:

1) Un criptoanálisis del generador auto-shriking con órdenes de complejidad que superan los alcanzados por otros autores hasta la fecha [9], [14].
2) Obtener resultados de complejidad en tiempo menores que $O(2^{0.5L})$, de una complejidad de datos menor que $O(2^{0.25L})$ y una complejidad en memoria de $O(L^2)$.
3) La pre-computación es baja en tiempo. Debido a la disminución de los órdenes de magnitud, específicamente la complejidad en tiempo, este criptoanálisis propuesto es concebible en tiempos computacionales aceptables, a diferencia de los propuestos por otros autores.
4) Exponer claramente las mejoras logradas con respecto a otros autores, profundizando en el estudio y logrando una originalidad en el criptoanálisis basada en una selección fina de cada una de las suposiciones que intervienen en el proceso.
5) Cumplir el objetivo planteado en el proyecto Stork [12] de romper el generador auto-shrinking con complejidades menores a las logradas mediante la técnica de Time Memory Trade Off.
6) Dejar planteado, para un trabajo futuro, un esquema razonable y factible de desarrollo en hardware, que permita en tiempos computacionales aceptables, romper el generador auto-shrinking.

AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente por el CDTI y las empresas INDRA, Unión Fenosa, Tecnobit, Visual Tools, Brainstorm, SAC y Technosafe en el marco del Proyecto Cenit-HESPERIA; también ha sido financiado por el Ministerio de Ciencia e Innovación y el fondo europeo FEDER en el marco del Proyecto TIN2008-02236/TSI.

REFERENCES

[1] Bluetooth, Specifications of the Bluetooth system, Version 1.1, 2001, available at http://www.bluetooth.com/Rivest, L.: RSA Data Security, Inc., March 12, 1998.
[2] A. Fúster-Sabater and P. Caballero-Gil, Strategic Attack on the Shrinking Generator, Theoretical Computer Science, Vol. 409, No. 3, pp. 530-536, December 2008.
[3] A. Fúster-Sabater and P. Caballero-Gil, Cryptanalytic Attack on Cryptographic Sequence Generators: The Class of Clock-Controlled Shrinking Generators. Proc. of ICCSA 2008, Part II. Lecture Notes on Computer Science, Springer-Verlag, Vol. 5073, pp. 668-679, 2008.
[4] S.W. Golomb, Shift Register-Sequences, Aegean Park Press, Laguna Hill, 1982.
[5] GSM, Global Systems for Mobile Communications, available at http://cryptome.org/gsm-a512.htm
[6] M. Hellman, A Cryptanalytic Time-Memory Trade-Off, IEEE Trans. Informat. Theory, Vol. 26, No. 4, pp. 234-247, 1980.
[7] W. Meier and O. Staffelbach, The Self-Shrinking Generator, in Proc. EUROCRYPT94. Lecture Notes in Computer Science, Springer Verlag, Vol. 950, pp. 205-214, 1995.
[8] A.J. Menezes *et al.*, Handbook of Applied Cryptography, New York:CRC Press, 1997.
[9] M.J. Mihaljevic, A Faster Cryptanalysis of the Self-Shrinking Generator, in Proc. ACISP96. Lecture Notes in Computer Science, Springer Verlag, Vol. 1172, pp. 182-189, 1996.
[10] National Instruments, http://www.ni.com/pdf/products/us/cat-flexriofpga.pdf
[11] RSA Data Security, Inc., Internal report, 12, 1999.
[12] Stork Proyect, available at http://www.stork.eu.org/documents/RUB-D6-2-1.pdf

[13] Xilinx, http://www.xilinx.com. 11. National Instruments, available at http://www.ni.com/pdf/products/us/cat-flexriofpga.pdf
[14] B. Zhang and D. Feng, New Guess-and-Determine Attack on the Self-Shrinking Generator, in Proc. ASIACRYPT06. Lecture Notes in Computer Science, Springer Verlag, Vol. 4284, pp. 54-68, 2006.

Paralelización del algoritmo Rho de Pollard con requisitos de memoria negligibles

Francesc Sebé, Jordi Pujolàs, Teodoro Lairla
Departamento de Matemática
Universitat de Lleida
C/ Jaume II, 69. E-25001 Lleida
Email: {fsebe,jpujolas,teo}@matematica.udl.cat

***Resumen*—El algoritmo Rho de Pollard es el mejor algoritmo conocido para resolver el problema del logaritmo discreto en algunos grupos como el grupo de puntos de una curva elíptica o la variedad jacobiana de una curva hiperelíptica. El algoritmo se basa en hallar una colisión dentro de una secuencia pseudoaleatoria que una vez encontrada permite calcular la solución del problema. Exiten diversos métodos para hallar esta colisión que pueden clasificarse según si su uso de memoria es negligible o no. En este artículo presentamos una propuesta para paralelizar la búsqueda de colisiones con uso de memoria negligible que ofrece un *speedup* proporcional al número de procesadores disponibles.**

I. Introducción

La seguridad de las cifras de clave pública [13] se basa en la supuesta inexistencia de algoritmos eficientes para resolver determinados problemas matemáticos tales como el problema de la factorización entera (cifra RSA [8]) o el problema del logaritmo discreto (cifra ElGamal [2]). A la hora de fijar los parámetros de un criptosistema basado en alguno de estos problemas es necesario tener en cuenta el coste del mejor algoritmo que lo resuelve y escoger así unos parámetros lo bastante grandes para garantizar la seguridad pero sin sobredimensionar el sistema ya que esto penalizaría el rendimiento.

Sea G un grupo multiplicativo de orden p y sea g un generador de G. Dado $h \in G$, el problema del logaritmo discreto consiste en encontrar un entero m tal que $g^m = h$ (esta expresión puede ser reescrita como $m = \log_g h$). Cuando el orden del grupo G tiene un factor primo grande, existen grupos donde el mejor algoritmo conocido para su resolución tiene un coste no polinomial, haciéndolos adecuados para su uso en criptografía.

Por ejemplo, cuando G es el subgrupo multiplicativo de un cuerpo de Galois $\mathbb{F}_q^*$, el algoritmo *index-calculus* calcula un logaritmo discreto con un coste subexponencial $O(e^{\sqrt{\log q \log\log q}})$. Si q es primo, el algoritmo *number field sieve* reduce este coste a $O(e^{1{,}923(\log q)^{\frac{1}{3}}(\log\log q)^{\frac{2}{3}}})$, véase [10].

Para otros grupos, tales como el grupo de puntos de una curva elíptica o la variedad jacobiana de una curva hiperelíptica, el mejor algoritmo conocido es el algoritmo *Rho de Pollard* [7] cuyo coste (exponencial respecto a la longitud de p) es $O(\sqrt{p})$, donde p es el cardinal del grupo G. Su coste exponencial implica que su ejecución termina en un tiempo aceptable solamente cuando p tiene una longitud reducida. Para resolver instancias con longitudes mayores de p será necesaria la utilización de varios procesadores ejecutando una versión paralelizada del algoritmo.

La calidad de una paralelización se mide mediante el parámetro *speedup* que se define como el cociente entre el tiempo de ejecución de un algoritmo y el tiempo de ejecución de su versión paralelizada. El valor óptimo de este parámetro se obtiene cuando corresponde con el número de procesadores de que disponemos. Alcanzar el valor óptimo significa que si disponemos de M procesadores estamos dividiendo el tiempo de ejecución por M. Este valor óptimo no suele alcanzarse ya que un algoritmo paralelizado pierde rendimiento a causa de la gestión de procesos y por el intercambio de mensajes e información que hay entre ellos.

En este artículo presentamos una forma de paralelizar el algoritmo Rho de Pollard mediante la cual se consigue un *speedup* proporcional a M y cuyos requisitos de memoria son negligibles.

II. El algoritmo Rho de Pollard

El algoritmo Rho de Pollard resuelve el problema del logaritmo discreto sobre un grupo G de orden p (dados $g, h \in G$, calcula $\log_g h$) a partir de dos pares distintos (a_i, b_i), (a_j, b_j) de enteros módulo p que satisfacen

$$g^{a_i} h^{b_i} = g^{a_j} h^{b_j}.$$

Operando la expresión anterior se obtiene,

$$h^{b_i - b_j} = g^{a_j - a_i}$$

lo cual permite calcular $\log_g h$ como,

$$\log_g h = (a_j - a_i)(b_i - b_j)^{-1} \pmod{p}. \qquad (1)$$

El algoritmo Rho de Pollard utiliza una función $f : G \to G$ tal que dada una tripleta (x_i, a_i, b_i) que satisface $x_i = g^{a_i} h^{b_i} \in G$, es fácil calcular otra tripleta distinta (x_j, a_j, b_j) tal que $x_j = f(x_i)$ y $x_j = g^{a_j} h^{b_j}$. La función f ha de tener un comportamiento pseudoaleatorio. Pollard sugiere dividir los elementos de G en tres grupos distintos T_1, T_2 y T_3 y el uso de la siguiente función

$$f(x) = \begin{cases} gx, & \text{si } x \in T_1, \\ x^2, & \text{si } x \in T_2, \\ hx, & \text{si } x \in T_3. \end{cases}$$

En [11] se estudia el uso de otro tipo de funciones. De este modo se puede definir una función F que retorna $(x_j, a_j, b_j) = F(x_i, a_i, b_i)$, con $x_j = f(x_i)$, de la siguiente manera,

$$F(x_i, a_i, b_i) = \begin{cases} (gx_i,\ a_i + 1,\ b_i), & \text{si } x_i \in T_1, \\ (x_i^2,\ 2a_i,\ 2b_i), & \text{si } x_i \in T_2, \\ (hx_i,\ a_i,\ b_i + 1), & \text{si } x_i \in T_3, \end{cases}$$

siendo las operaciones sobre a_i y b_i módulo p.

Comenzando desde una tripleta cualquiera (x_0, a_0, b_0), se puede construir la secuencia $\{(x_i, a_i, b_i)\}_{i \geq 0}$ donde $(x_i, a_i, b_i) = F(x_{i-1}, a_{i-1}, b_{i-1})$, para $i \geq 1$. Dado que el grupo G es finito, existe un índice t para el cual $x_t = x_{t-s}$ para algún $s \geq 0$. Entonces $x_i = x_{i-s}$ para todo $i \geq t$ con lo cual la secuencia entra en un ciclo. A partir de la paradoja del aniversario se demuestra que el camino generado antes de que se produzca la primera colisión (valor del índice t) tendrá una longitud esperada de $\sqrt{\pi p/2}$. El algoritmo Rho de Pollard encuentra una de estas colisiones (dos tripletas distintas (x_i, a_i, b_i), (x_j, a_j, b_j) con $x_i = x_j$) y resuelve el logaritmo discreto aplicando la fórmula (1). Dicha colisión puede buscarse mediante el algoritmo de Floyd (tal como se propone en [7]) o utilizando el algoritmo de Brent [1].

II-A. Algoritmo de Floyd para búsqueda de colisiones

Para encontrar una de estas colisiones puede usarse el algoritmo de búsqueda de ciclos de Floyd (también conocido como el algoritmo de "la tortuga y la liebre"). Este algoritmo utiliza dos tripletas que avanzan a lo largo de una misma secuencia que comienza en una tripleta cualquiera (x_0, a_0, b_0). Una de ellas, la tortuga, avanza un paso en cada iteración mientras que la otra, la liebre, avanza dos pasos por iteración. Una vez las dos tripletas han entrado en el ciclo, hay un momento en el que la liebre alcanza a la tortuga, hallándose de este modo una colisión.

Algorithm 1 Algoritmo de Floyd

Input: Una tripleta (x_0, a_0, b_0)
Output: Dos tripletas distintas (x_T, a_T, b_T), (x_L, a_L, b_L) con $x_T = x_L$

$(x_T, a_T, b_T) := (x_0, a_0, b_0)$;
$(x_L, a_L, b_L) := F(x_T, a_T, b_T)$;
while $x_L \neq x_T$ **do**
    $(x_T, a_T, b_T) := F(x_T, a_T, b_T)$;
    $(x_L, a_L, b_L) := F(F(x_L, a_L, b_L))$;
end
return (x_T, a_T, b_T), (x_L, a_L, b_L);

Este algoritmo solamente necesita memoria para almacenar dos tripletas. El número de operaciones a realizar sobre el grupo para encontrar una colisión es aproximadamente $3\sqrt{p}$.

II-B. Algoritmo de Brent para búsqueda de colisiones

El algoritmo de Brent [1] permite encontrar colisiones mediante una técnica distinta también basada en el uso de dos tripletas llamadas tortuga y liebre. En este algoritmo la liebre avanza un paso en cada iteración mientras que la tortuga permanece quieta. La tortuga toma el valor de la libre una vez esta última ha realizado 2^i saltos desde la última actualización de la tortuga. El índice i toma un valor inicial de 1 y es incrementado en una unidad cada vez que la tortuga es actualizada.

Algorithm 2 Algoritmo de Brent

Input: Una tripleta (x_0, a_0, b_0)
Output: Dos tripletas distintas (x_T, a_T, b_T), (x_L, a_L, b_L) con $x_T = x_L$

$(x_T, a_T, b_T) := (x_0, a_0, b_0)$;
$(x_L, a_L, b_L) := F(x_T, a_T, b_T)$;
$Longitud := Pasos := 1$;
while $x_L \neq x_T$ **do**
    if *Pasos=Longitud* **then**
        $(x_T, a_T, b_T) := (x_L, a_L, b_L)$;
        $Longitud := 2 * Longitud$;
        $Pasos := 0$;
    end
    $(x_L, a_L, b_L) := F(x_L, a_L, b_L)$;
    $Pasos := Pasos + 1$;
end
return (x_T, a_T, b_T), (x_H, a_H, b_H);

Igual que el algoritmo de Floyd, este algoritmo tiene un coste espacial negligible. En [1] se afirma que su utilización incrementa la velocidad del algoritmo de Pollard alrededor de un 24 %.

II-C. Búsqueda de colisiones con uso de memoria

Tanto el algoritmo de Floyd como el de Brent encuentran una colisión sin apenas utilizar memoria con aproximadamente $3\sqrt{p}$ operaciones sobre el grupo. Existen propuestas que reducen el tiempo necesario para encontrar la colisión a expensas de aumentar los requisitos de uso de memoria. Por ejemplo, en [9] se da una propuesta que reduce el número de operaciones a realizar sobre el grupo a $\sqrt{p}\left(1 + \frac{1}{\sqrt{M}}\right)$ si se dispone de memoria para almacenar M tripletas. Otro ejemplo es la propuesta de [4] donde se usa una pila de memoria de tamaño $O(\log \sqrt{p})$.

III. Paralelización del algoritmo Rho de Pollard

El algoritmo Rho de Pollard ejecutado sobre un grupo G de orden p tiene un coste $O(\sqrt{p})$. Por tanto, a medida que p aumenta de tamaño, el tiempo necesario para ejecutarlo aumenta exponencialmente hasta convertirse en un problema intratable. El rango tamaños de p para los cuales el algoritmo es tratable puede aumentarse cuando disponemos de diversos procesadores mediante la ejecución de una versión paralelizada del algoritmo.

III-A. *Paralelización directa*

Cuando se dispone de varios procesadores, el algoritmo Rho de Pollard puede paralelizarse de forma trivial haciendo que cada procesador ejecute el algoritmo de forma independiente partiendo de una tripleta inicial distinta. El algoritmo paralelizado se detiene cuando cualquiera de los procesadores halla la primera colisión. Si disponemos de M procesadores y buscamos la colisión mediante el algoritmo de Floyd o el de Brent (lo cual nos permite un uso negligible de memoria), el número esperado de operaciones sobre el grupo realizadas por cada procesador en el momento de encontrarse la primera colisión es de $3\sqrt{\frac{p}{M}}$ con lo cual la propuesta proporciona un *speedup* que es solamente un factor de $\sqrt{M}$ lo cual resulta muy poco eficiente (por ejemplo, para reducir el tiempo de cálculo a una cuarta parte es necesario el uso de 16 procesadores).

III-B. *Búsqueda paralela de colisiones mediante tripletas distinguidas*

En [6] se propone una versión paralela de búsqueda de colisiones que permite paralelizar el algoritmo Rho de Pollard con un *speedup* proporcional a M cuando se dispone de M procesadores y de una zona de memoria para almacenar tripletas.

En esta propuesta, cada procesador escoge una tripleta al azar (x_0, a_0, b_0) desde la cual construye un camino de tripletas $(x_i, a_i, b_i) = F(x_{i-1}, a_{i-1}, b_{i-1})$, $i > 0$, hasta encontrar un valor x_d que satisface una determinada condición de fácil comprobación. Esta tripleta distinguida, (x_d, a_d, b_d) es enviada a un procesador encargado de recoger y almacenar las tripletas distinguidas que le irán mandando los otros procesadores. El procesador que halló la tripleta distinguida, después de mandarla, generará una nueva tripleta inicial al azar y repetirá el mismo proceso. El algoritmo termina cuando el procesador encargado de recoger las tripletas distinguidas recibe una tripleta (x_d, a_d, b_d) cuyo valor x_d coincide con el de otra tripleta distinta almacenada previamente. En este momento ya se ha hallado una colisión y el algoritmo termina.

La propiedad que distingue algunos elementos del grupo G ha de escogerse de manera que la proporción de elementos que la satisfacen sea θ. Dado θ, el número de operaciones sobre el grupo realizadas por cada procesador antes de hallarse la primera colisión (coste temporal) es

$$\frac{1}{M}\sqrt{\frac{\pi p}{2}} + \frac{1}{\theta} \tag{2}$$

Puede ocurrir que la secuencia calculada por alguno de los procesadores entre en un ciclo sin puntos distinguidos. Una forma de recuperarse de esta situación consiste en establecer una longitud de camino máxima (en [6] se recomienda $20/\theta$) que si es alcanzada por algún camino, éste será abandonado y se iniciará el recorrido de otro camino desde una tripleta aleatoria generada al azar.

La fórmula 2 supone que hay suficiente memoria para almacenar todas las tripletas enviadas al procesador que las almacena. Como tenemos M procesadores trabajando en paralelo y cada uno de ellos envía una proporción θ de tripletas a la lista, el número de tripletas que serán almacenadas es

$$\theta\sqrt{\frac{\pi p}{2}} + M. \tag{3}$$

El parámetro θ ajusta el compromiso entre los costes temporal y espacial. Un valor grande de θ reduce el coste temporal a expensas de aumentar el almacenamiento y el número de tripletas enviadas (cosa que podría llegar a causar un cuello de botella).

Cuando la memoria disponible es limitada tenemos dos opciones. La primera consiste en tomar el valor más grande posible de θ tal que el almacenamiento previsto necesario (expresión 3) corresponda con la memoria disponible. La segunda opción (sugerida en [6]) es la de tomar un θ más grande y añadir tripletas a la lista hasta que la memoria se agote. A partir de este momento, cada vez que se deba guardar una tripleta nueva, se eliminará alguna de las almacenadas previamente.

Esta propuesta proporciona un *speedup* que al aumentar linealmente con el número de procesadores resulta eficiente. Por otro lado, el almacenamiento de tripletas distinguidas obliga a disponer de un espacio de memoria cuyo tamaño repercute en el coste temporal del algoritmo (a mayor tamaño, menor es el tiempo de ejecución).

IV. Nueva propuesta

En este trabajo se presenta una nueva forma de paralelizar el algoritmo Rho de Pollard con *speedup* proporcional al número de procesadores y cuyos requisitos de memoria son muy reducidos.

En esta nueva propuesta, cada uno de los procesadores ejecuta el algoritmo de Pollard buscando la colisión de dos tripletas mediante el algoritmo de Brent tomando una tripleta al azar como punto de partida de su secuencia. Sin añadir nada más, esto sería una simple paralelización trivial con la cual se obtendría un *speedup* proporcional a $\sqrt{M}$.

Para aumentar el *speedup* y hacerlo proporcional a M nuestra propuesta modifica la paralelización trivial haciendo que en cada iteración, un procesador comprueba, no solamente si su liebre ha alcanzado a su tortuga, sino si su liebre ha alcanzado a su propia tortuga o a la de cualquier otro de los procesadores.

En el algoritmo 3 se da una descripción formal de la propuesta suponiendo que se dispone de un espacio de memoria compartido entre todos los procesadores donde cada uno de ellos tiene una celda reservada donde almacena el valor de su tripleta "tortuga". El uso de este espacio de memoria compartido permite que esta paralelización tenga los mismos requisitos de memoria que la paralelización trivial. Para una implementación eficiente, este espacio de memoria debe ser implementado utilizando una estructura de datos de acceso rápido como por ejemplo un *hash* [12].

Algorithm 3 Nueva propuesta para la paralelización del algoritmo Rho de Pollard

Input: Un grupo G de orden p y dos elementos $g, h \in G$
Output: El logaritmo discreto $m = \log_g h$

Cada Procesador i **hace:**
  Escoger $a_L, b_L \in [0, p-1]$ al azar;
  Calcular $x_L = g^{a_L} h^{b_L}$;
  Almacenar (x_L, a_L, b_L) en la celda i de memoria;
  $(x_L, a_L, b_L) := F(x_L, a_L, b_L)$;
  $Longitud := Pasos := 1$;
  while *No haya una tripleta* (x_j, a_j, b_j) *con* $x_j = x_L$ *en memoria* **do**
    if *Pasos=Longitud* **then**
      Almacenar (x_L, a_L, b_L) en la celda i de memoria;
      $Longitud := 2 * Longitud$;
      $Pasos := 0$;
    end
    $(x_L, a_L, b_L) = F(x_L, a_L, b_L)$;
    $Pasos := Pasos + 1$;
  end
  Recuperar la tripleta (x_T, a_T, b_T) con $x_T = x_L$ de memoria;
  Ordenar a todos los procesadores que se detengan;
  Calcular $m := (a_T - a_L)(b_L - b_T)^{-1} \pmod p$;
  return m;

V. Resultados experimentales

El algoritmo 3 ha sido implementado en C++ para resolver un logaritmo discreto planteado sobre un subgrupo de orden primo del grupo multiplicativo $\mathbb{F}_q^*$, con q primo, utilizando la librería de números grandes NTL [5]. Dado que el algoritmo es de tipo heurístico, cada experimento se ha repetido veinte veces, variando al azar las tripletas de inicio, y se ha calculado el tiempo medio de ejecución.

V-A. Resultados en un multicomputador de memoria compartida

A continuación se presentan los resultados obtenidos en una implementación para un multicomputador de memoria compartida formado por dos procesadores Intel Xeon de 3.16 GHz de cuatro núcleos cada uno (disponiendo así de un total de ocho núcleos de cálculo). La zona de memoria donde se almacenan las tortugas se ha implementado sobre un segmento de memoria compartida. En el cuadro I se muestran los resultados.

Para cuantificar el descenso en el tiempo de ejecución, se han calculado los *speedup* que se muestran en el Cuadro II. La obtención, en algunos casos, de valores de *speedup* superiores al número de procesadores se debe a que estamos trabajando con un algoritmo cuyo tiempo de ejecución es aleatorio y con una varianza muy elevada (en algunos casos, para la resolución del mismo problema se han obtenido tiempos tan distintos como 362 y 1570 segundos). La conclusión importante, más que los valores concretos de *speedup* obtenidos, es su tendencia a aumentar linealmente respecto al número de procesadores utilizado.

	Número de procesadores				
Bits	1	2	4	6	8
52	189	87	31	19	17
56	1119	438	194	136	81
60	1101	722	437	297	269
64	13210	5281	1906	1600	741

Cuadro I
TIEMPO MEDIO (EN SEGUNDOS) NECESARIO PARA RESOLVER UN LOGARITMO DISCRETO SOBRE UN GRUPO MULTIPLICATIVO $\mathbb{F}_q^*$, CON q PRIMO DE 52, 56, 60 Y 64 BITS DE LONGITUD EN UN MULTICOMPUTADOR DE MEMORIA COMPARTIDA.

	Número de procesadores			
Bits	2	4	6	8
52	2.2	6.1	10	11.1
56	2.5	5.8	8.2	13.9
60	1.5	2.5	3.7	4.1
64	2.5	7	8.3	17.8

Cuadro II
SPEEDUP DEL ALGORITMO PARALELIZADO EN UN MULTICOMPUTADOR DE MEMORIA COMPARTIDA.

V-B. Resultados en un multicomputador de memoria distribuida

Los mismos experimentos se han llevado a cabo en un *cluster* formado por ocho nodos con procesadores Intel Core 2 Quad de 2.4 GHz interconectados a través de una red Gigabit Ethernet. La implementación para esta plataforma se ha hecho en C++ utilizando la librería de comunicaciones MPI [3]. La compartición de tripletas tortuga se ha implementado de tal forma que cada procesador dispone de una copia de todas las tripletas tortuga. Para mantener la coherencia de estos datos, cada vez que un procesador actualiza su tripleta tortuga, informa al resto de procesadores mediante el envío de un mensaje *broadcast*. Los tiempos medios y *speedup* obtenidos se hallan listados en los cuadros III y IV, respectivamente, y permiten reafirmar las conclusiones obtenidas en el primer experimento. En este entorno, la paralelización presenta un rendimiento (*speedup*) menor debido a la penalización que conlleva que la comunicación entre procesos deba realizarse mediante envío de mensajes.

	Número de procesadores				
Bits	1	2	4	6	8
52	240	159	51	39	37
56	1085	697	288	245	284
60	1967	1093	801	626	551
64	16884	7571	5707	3132	2139

Cuadro III
TIEMPO MEDIO (EN SEGUNDOS) NECESARIO PARA RESOLVER UN LOGARITMO DISCRETO SOBRE UN GRUPO MULTIPLICATIVO $\mathbb{F}_q^*$, CON q PRIMO DE 52, 56, 60 Y 64 BITS DE LONGITUD EN UN ENTORNO MPI.

	Número de procesadores			
Bits	2	4	6	8
52	1.5	4.7	6.2	6.5
56	1.6	3.8	4.4	3.8
60	1.8	2.5	3.1	3.6
64	2.2	3	5.4	7.9

Cuadro IV
SPEEDUP DEL ALGORITMO PARALELIZADO EN UN ENTORNO MPI.

VI. CONCLUSIONES Y TRABAJO FUTURO

En este artículo se ha presentado una nueva forma de paralelizar el algoritmo Rho de Pollard con la cual se obtiene un *speedup* proporcional al número de procesadores utilizados y cuyos requisitos de memoria son los mismos que en una paralelización trivial.

Como trabajo futuro se plantea el estudio del rendimiento de este método en entornos heterogéneos donde los computadores se encuentran distribuidos geográficamente e interconectados a través de Internet (*grid computing*).

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia mediante los proyectos MTM2007-66842-C02-02 y "ARES" CSD2007-0004 y por la Generalitat de Catalunya mediante el proyecto 2009SGR-442.

REFERENCIAS

[1] R.P. Brent, "An improved Monte Carlo factorization algorithm", *BIT*, vol. 20, pp. 176–184, 1980.
[2] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithm", *IEEE Trans. Information Theory*, vol. 31, pp. 469–472, 1985.
[3] LAM/MPI Parallel computing. `http://www.lam-mpi.org`.
[4] G. Nivasch, "Cycle detection using a stack", *Inf. Proc. Letters*, vol. 90, pp. 135–140, 2004.
[5] NTL: A Library for doing Number Theory. `http://www.shoup.net`.
[6] P. van Oorschot, M.J. Wiener, "Parallel collision search with cryptanalytic applications", *Journal of Cryptology*, 12, pp. 1–28, 1999.
[7] J.M. Pollard, "Monte Carlo methods for index computation (mod p)", *Math.Comp.*, vol. 32 (143), pp. 918–924, 1978.
[8] R. Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM* 21 (2), pp. 120-126, 1978.
[9] R. Sedgewick, T.G. Szymanski, C. Yao, "The complexity of finding cycles in periodic functions", *Siam J. Comput.* 11 (2), pp. 376–390, 1982.
[10] C. Studholme, "The discrete log problem", Manuscript, 2002.
[11] E. Teske, "Speeding up Pollard's Rho method for computing discrete logarithms", *Lecture Notes in Computer Science*, vol. 1423, pp. 541–554, 1998.
[12] A. Drozdek, *Data structures and algorithms in C++*, Thomson learning, 2005.
[13] H.C.A. van Tilborg, *Fundamentals of cryptology*, Kluwer Academic Publishers, 2000.

Taxonomía de ataques a entornos de creación de firmas electrónicas

Jorge López Hernández-Ardieta Ana Isabel González-Tablas Ferreres Benjamín Ramos Álvarez
Grupo SeTI
Departamento de Informática
Universidad Carlos III de Madrid
Email: jlhernan,aigonzal,benja1@inf.uc3m.es

***Resumen*—La firma electrónica se ha convertido en un elemento fundamental en la sociedad de la información, disfrutando del reconocimiento como medio de prueba en procedimientos legales. Así mismo, la firma electrónica es considerada una evidencia de no repudio respecto a los compromisos adquiridos por los participantes en una transacción electrónica. Sin embargo, la realidad nos muestra que la tecnología no está libre de vulnerabilidades, existiendo múltiples amenazas que pueden comprometer la seguridad de los sistemas. Es por tanto imprescindible disponer de las herramientas adecuadas que asistan al desarrollo de sistemas seguros, de forma que pueda aplicarse la legislación con plenas garantías para las partes implicadas. En particular, el proceso de generación de la firma es una de las etapas más sensibles, y a la cual se debe prestar especial atención. En este artículo se presenta la primera taxonomía completa de ataques a entornos de creación de firma, y que permitirá el análisis riguroso y sistemático de las causas que pueden socavar la fiabilidad de la misma, y así poder actuar en consecuencia.**

I. Introducción

Una taxonomía es un sistema o esquema para la clasificación sistemática del conocimiento. Mediante un análisis riguroso y sistemático, dicho conocimiento es clasificado en un conjunto limitado de categorías bien definidas. De esta forma, una taxonomía permite descomponer conceptos o fenómenos complejos en unidades de información más abordables, no sólo permitiendo su estudio sino también proporcionando una base de partida común sobre la cual analizar y clasificar hechos desconocidos hasta el momento.

Para que una taxonomía sea útil, debe estar muy especializada en un área de conocimiento o fenómeno concreto. Hasta la fecha se ha propuesto un gran número de taxonomías centradas en los sistemas de la información, así como taxonomías focalizadas en el área de la seguridad. Existen taxonomías para la clasificación de sistemas de detección de intrusiones [11], errores y vulnerabilidades en los sistemas [12], [13], [14], ataques software [16], incidentes de seguridad en redes de comunicaciones [15], [17] o ataques a dispositivos seguros [21]. Estas taxonomías han permitido profundizar en el estudio de múltiples problemáticas de seguridad mediante la categorización de ataques y vulnerabilidades.

En cuanto a taxonomías relacionadas con firma electrónica, Kain propuso en 2003 una taxonomía no formal de ataques a firmas electrónicas mediante la manipulación del documento a firmar [19]. Sin embargo, la taxonomía no es completa, y se centra exclusivamente en ataques que modifican la semántica del documento, obviando otros muchos tipos de ataque posibles. Otros investigadores han estudiado vulnerabilidades y ataques a tarjetas inteligentes [20], cuyo compromiso claramente afecta a la fiabilidad de la firma electrónica, aunque sólo unos pocos han proporcionado una clasificación rigurosa de ataques a dichos dispositivos [21]. Por otra parte, numerosos trabajos se han centrado en los ataques de canal indirecto, los cuales tienen un gran impacto en la seguridad de los dispositivos criptográficos [22].

Sin embargo, a día de hoy no se ha propuesto ninguna taxonomía que abarque de forma integral y rigurosa los problemas de seguridad que afectan a la firma electrónica. El presente artículo pretende dar respuesta a esta necesidad, ya que las firmas electrónicas se han convertido en un elemento clave en múltiples escenarios, pero todavía no se ha estudiado de forma sistemática las amenazas que pueden influir en la fiabilidad de las mismas. Deseamos que la taxonomía aquí propuesta sirva de guía para diseñar soluciones de firma más seguras y robustas, una vez conocidos los riesgos existentes.

El artículo recoge en primer lugar la motivación en la Sección II. Posteriormente, la Sección III acota el modelo de sistema que tomaremos como objeto de estudio para definir la taxonomía. La Sección IV propone la primera taxonomía completa de ataques a entornos de creación de firma. La evaluación de la taxonomía frente a los requisitos fundamentales para taxonomías se incluye en la Sección V. Por último, se concluye el artículo en la Sección VI.

II. Motivación

La firma electrónica se ha convertido en un elemento fundamental en la sociedad de la información. Numerosas legislaciones de todo el mundo han otorgado a la firma electrónica una equivalencia funcional respecto a la firma manuscrita, así como un reconocimiento como mecanismo de seguridad primordial en transacciones electrónicas, especialmente dentro del ámbito del comercio electrónico [2], [3], [4], [5], [6].

En el caso particular de las legislaciones Europea y Española, la firma electrónica disfruta de plena eficacia jurídica, pudiendo ser empleada como medio de prueba en procedimientos legales [7]. Dependiendo de los requisitos cumplidos por la firma en cuestión, ésta contará con un juicio favorable de validez a priori (*ex ante*) o necesitará de un juicio a posteriori (*ex post*) del Tribunal. En el primer caso, si el supuesto

firmante repudiara el compromiso adquirido en un documento firmado, debería aportar evidencias en sentido contrario y que generaran una duda razonable respecto a la seguridad de la firma. Así pues, la carga de prueba en este escenario recae sobre el supuesto firmante que niega la validez de la firma [1]. En el segundo caso, el supuesto firmante podría también repudiar la autoría de dicha firma pero sería la otra parte afectada quien debiera aportar garantías sobre la seguridad de la misma.

En esta línea, existe también un reconocimiento en los estándares internacionales de proporcionar a las firmas electrónicas (criptográficas) la cualidad de evidencia de no repudio [8]. Una evidencia de no repudio es información que, bien por sí misma o usada en conjunción con otros datos, se emplea para probar la ocurrencia de un evento o acción. Aunque la evidencia no prueba necesariamente por sí misma la veracidad del hecho, sí es usada para tal fin. Por tanto, una evidencia de no repudio que se verifica correctamente de acuerdo a la política de no repudio que aplica es suficiente para resolver una posible disputa, impidiendo al firmante repudiar el compromiso adquirido en la transacción.

Así pues, consideramos imprescindible disponer de las herramientas adecuadas para el estudio riguroso del problema de seguridad de las firmas electrónicas, con el fin de poder asistir al desarrollo de sistemas más seguros que permitan aplicar la legislación con plenas garantías para las partes afectadas. En este sentido, una taxonomía de ataques permitirá conocer dicha problemática, y actuar en consecuencia. Dicha taxonomía no se ha propuesto hasta la fecha.

La taxonomía aquí presentada considera firmas electrónicas generadas mediante criptografía asimétrica (firmas digitales), al ser no sólo una de las posibles tecnologías concretas de implementación sino también la única tecnología que, a día de hoy, es capaz de alcanzar determinados requisitos establecidos en la legislación. En particular, la taxonomía se centra en una de las etapas más sensibles del ciclo de vida de una firma, esto es, la fase de generación.

III. Modelo de Sistema

La Figura 1 muestra el modelo de sistema que tomamos como punto de partida para el estudio de los ataques al entorno de creación de firmas. Dicho modelo se sustenta en el propuesto en el estándar CEN CWA 14170 [9].

La aplicación de creación de firma (SCA) es la aplicación dentro del sistema (SCS) que incorpora la funcionalidad de generación de firmas electrónicas, apoyándose en el dispositivo (seguro) de creación de firmas (S)SCDev. El entorno de creación de firmas (SCE) incluye los entornos físico y lógico del SCS, así como el firmante y las políticas existentes. El SCDev es el dispositivo software o hardware que incorpora los datos de creación de firma (SCD), es decir, la clave privada de firma usada por el firmante para generar las firmas electrónicas. El SSCDev es un SCDev que cumple con los requisitos estipulados en el Anexo III de la Directiva Europea de firma [2]. El acceso al SCD se protege mediante los datos

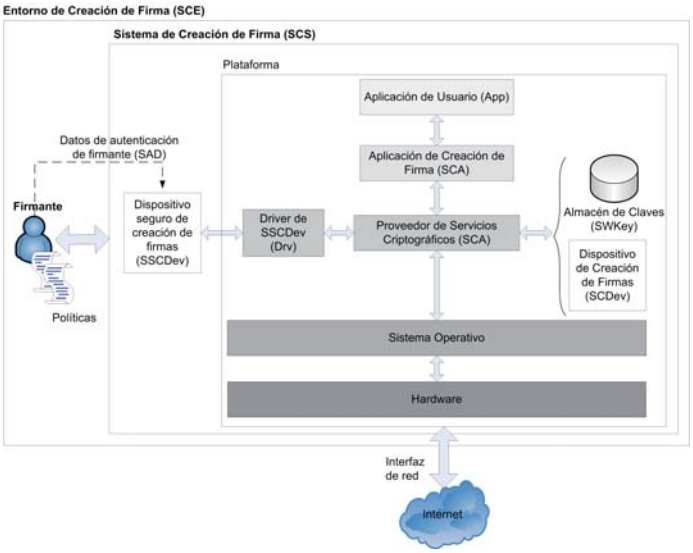


Figura 1. Modelo de Sistema

de autenticación del firmante (SAD), consistentes en un PIN o contraseña.

En este artículo nos referimos al término DTBS cuando los datos a firmar incorporan tanto el documento en sí como una serie de atributos de firma que enriquecen la semántica de dicho documento o particularizan el compromiso adquirido en el acto de firma. Por otra parte, DTBSR se referirá a la representación del DTBS que se envía al SCDev/SSCDev para el cómputo de la firma - generalmente se corresponderá con el resumen criptográfico del DTBS.

IV. Taxonomía de Ataques a Entornos de Creación de Firma

La taxonomía aquí propuesta toma como modelo de sistema el descrito en la Sección III. Así pues, la taxonomía categoriza ataques orientados a comprometer la seguridad del proceso de generación de firmas electrónicas llevado a cabo en un sistema que cumple el modelo indicado.

En particular, la taxonomía está formada por cuatro dimensiones, que se detallan en las siguientes subsecciones. El concepto de dimensión fue introducido por Landwehr et al. [12], y es una propiedad que permite la clasificación de un ataque o vulnerabilidad desde un punto de vista integral. Cada ataque se descompone en varias propiedades, y cada propiedad se emplea para clasificar el ataque desde una perspectiva diferente, todas complementarias entre sí.

Las categorías tienen un identificador asociado consistente en el número de dimensión en el que están englobadas (D) y el número de categoría o subcategoría según el orden jerárquico establecido (CAT).

IV-A. Dimensión Objetivo del Atacante

Esta dimensión cubre el objetivo final del atacante. Identificamos tres categorías dentro de esta dimensión:

D1-CAT1: Firma no consciente de datos

El atacante no hace uso directo de los datos de creación de firma pero desea engañar al firmante para que, de forma no consciente, firme ciertos datos.

D1-CAT2: Uso no autorizado de los datos de creación de firma (SCD)

El atacante desea emplear los datos de creación de firma sin el conocimiento ni consentimiento del firmante. Para ello, el atacante necesitará bien comprometer directamente los datos de creación de firma bien tener acceso a la operación de firma.

D1-CAT3: Sustitución/Modificación de datos firmados

El atacante desea sustituir o modificar parte o la totalidad de los datos firmados, pero una vez que la firma se ha realizado.

IV-B. Dimensión Método de Ataque

Esta dimensión incluye las categorías de método de ataque empleado por el atacante para conseguir uno de los objetivos anteriormente identificados. En concreto, se han diseñado cinco categorías de primer nivel, que se refinan a su vez en subcategorías, y así hasta un cuarto nivel de profundidad en algunos casos:

D2-CAT1: Manipulación del entorno

Esta categoría abarca los métodos que pretenden manipular el entorno (principalmente la Plataforma del SCS) con el fin de influenciar en los datos firmados, bien sea durante la generación de la firma o con posterioridad.

D2-CAT2: Modificación previa a la generación de la firma

Esta categoría contiene los métodos de ataque que intervienen antes de la generación de la firma, y cuyo objetivo es la modificación de los datos a firmar, bien sea de forma directa (modificación del propio contenido a firmar) o indirecta (los datos fraudulentos se incorporan por referencia desde los datos a firmar).

D2-CAT2.1: Modificación de documento. Esta subcategoría de métodos se refiere a las modificaciones realizadas en el documento a firmar.

D2-CAT2.1.1: Adición de contenido dinámico. Esta subcategoría de métodos implica la inclusión de contenido dinámico en el documento a firmar. De esta forma, la sintaxis del documento se mantiene intacta, y por tanto la integridad de la firma, mientras que la semántica variará de acuerdo al comportamiento del código dinámico.

D2-CAT2.1.1.1: Código oculto. El atacante inserta etiquetas especiales o campos ocultos en el documento a firmar. Este código oculto será traducido por determinados valores dependiendo de condiciones específicas que pueden ser controladas por el atacante.

D2-CAT2.1.1.2: Código activo. El atacante inserta código especial, como scripts o macros, en el documento a firmar. Este código se ejecutará durante la apertura o visualización del documento, pudiendo realizar ciertas operaciones como la modificación del contenido mostrado.

D2-CAT2.1.1.3: Vínculos. El atacante inserta vínculos en el documento a firmar que apuntan a contenido externo no controlado por el firmante. Una vez realizada la firma, el atacante puede manipular dicho contenido externo a su antojo.

D2-CAT2.1.2: Modificación de contenido. El atacante modifica el contenido del documento a firmar, pero sin incluir ningún tipo de contenido dinámico (p. e. modificación del propio texto del documento).

D2-CAT2.2: Modificación de atributo. Esta subcategoría de métodos se refiere a las modificaciones realizadas en los atributos a firmar.

D2-CAT2.2.1: Adición de contenido dinámico. Esta subcategoría de métodos implica la inclusión de contenido dinámico en los atributos a firmar. El comportamiento es igual que en el caso anterior para el documento, pero enfocado en los atributos.

D2-CAT2.2.1.1: Código oculto. El atacante inserta etiquetas especiales o campos ocultos en los atributos a firmar. Este código oculto será traducido por determinados valores dependiendo de condiciones específicas que pueden ser controladas por el atacante.

D2-CAT2.2.1.2: Código activo. El atacante inserta código especial, como scripts o macros, en los atributos a firmar. Este código se ejecutará durante la visualización o aplicación de los atributos, pudiendo realizar ciertas operaciones como la modificación del contenido mostrado.

D2-CAT2.2.1.3: Vínculos. El atacante inserta vínculos en los atributos a firmar que apuntan a contenido externo no controlado por el firmante. Una vez realizada la firma, el atacante puede manipular dicho contenido externo a su antojo.

D2-CAT2.2.2: Modificación de contenido. El atacante modifica el contenido de los atributos a firmar, pero sin incluir ningún tipo de contenido dinámico (p. e. modificación del valor de cierto atributos).

D2-CAT2.3: Modificación de DTBS. El atacante modifica la información contenida en la estructura que representa los datos a firmar (DTBS).

D2-CAT2.4: Modificación de DTBSR. El atacante modifica el resumen de los datos a firmar (DTBSR). Este ataque se produciría en la última etapa antes de la computación de firma.

D2-CAT3: Modificación posterior a la generación de la firma

Esta categoría contiene métodos de ataque ejecutados una vez que la firma ha sido realizada, y cuyo objetivo es la modificación de los datos ya firmados, bien sea de forma directa (modificación del propio contenido firmado) o indirecta (modificación de datos referenciados desde los datos firmados).

D2-CAT3.1: Contenido externo. El atacante modifica datos externos referenciados desde los datos firmados (p. e. XSD, DTD). La diferencia entre este método y *D2-CAT2.1.1.3: Vínculos* o *D2-CAT2.2.1.3: Vínculos* radica en que, en este método, el vínculo a los datos externos no se incluye por el atacante, mientras que en estos otros dos casos, es el atacante

quien, de forma explícita, inserta dicho vínculo al contenido externo.

D2-CAT3.2: Criptoanálisis. El atacante aplica métodos de criptoanálisis para obtener un documento o atributos diferentes a los firmados de forma que no se invalide la firma.

D2-CAT3.2.1: Función resumen. El atacante aplica métodos específicamente diseñados para romper la seguridad de la función resumen empleada en el cálculo de la firma.

D2-CAT3.2.1.1: Ataque de colisión. El atacante es capaz de encontrar un par de mensajes $M \neq M'$ donde $hash(M) = hash(M')$ con una complejidad menor a $O(2^{n/2})$ (p. e. ataque del cumpleaños).

D2-CAT3.2.1.2: Ataque de preimagen. El atacante, dado un valor resumen H, es capaz de encontrar un mensaje M' donde $hash(M') = H$ con una complejidad menor que $O(2^n)$.

D2-CAT3.2.1.3: Ataque de segunda preimagen. El atacante, dado un mensaje M, es capaz de encontrar un segundo mensaje M', $M' \neq M$, que satisfaga $hash(M) = hash(M')$ con una complejidad menor que $O(2^n)$.

D2-CAT4: Invocación no autorizada de la operación de firma

Esta categoría recoge los métodos de ataque que no posibilitan al atacante conocer el SCD pero sí hacer uso de él sin el conocimiento ni consentimiento del usuario.

D2-CAT4.1: Compromiso de datos de autenticación de firmante (SAD). Esta subcategoría cubre los métodos que permiten al atacante recuperar el SAD.

D2-CAT4.1.1: Ingeniería social. El atacante manipula o engaña al firmante para que éste revele el SAD.

D2-CAT4.1.2: Intercepción del SAD. El atacante intercepta el SAD durante la operación del SCS.

D2-CAT4.1.2.1: Observación. El atacante observa al firmante introducir el SAD en la Plataforma.

D2-CAT4.1.2.2: Intercepción de comunicación entre procesos/entidades. El atacante intercepta el SAD durante su transmisión entre dos procesos o entidades físicos o lógicos que pertenecen al SCS.

D2-CAT4.1.2.3: Compromiso de extremo. Mediante el compromiso de la seguridad de un extremo (proceso o entidad que interviene en la comunicación del SAD dentro del SCS), el atacante es capaz de interceptar el SAD.

D2-CAT4.1.3: Obtención por prueba y error. El atacante emplea métodos probabilísticos (p. e. ataque de diccionario), técnicas de análisis de emanaciones acústicas del teclado o simplemente un ataque por fuerza bruta para adivinar el SAD.

D2-CAT4.2: Evitación de Autenticación. El atacante evita la autenticación para acceder a la operación de firma.

D2-CAT5: Compromiso de datos de creación de firma (SCD)

Esta categoría incluye métodos que permiten al atacante obtener el SCD. Esta es la categoría que aglutina los métodos de ataque más peligrosos, dado que el atacante podría, una vez conocido el SCD, realizar tantas firmas en nombre del usuario legítimo como deseara, incluso desde un entorno diferente al SCE.

D2-CAT5.1: Intercepción del SCD. El atacante intercepta el SCD durante el proceso de creación o distribución.

D2-CAT5.1.1: Intercepción de comunicación entre procesos/entidades. El atacante intercepta el SCD durante su transmisión entre dos procesos o entidades físicos o lógicos que pertenecen al SCS.

D2-CAT5.1.2: Compromiso de extremo. Mediante el compromiso de la seguridad de un extremo (proceso o entidad que interviene en la comunicación del SCD dentro del SCS), el atacante es capaz de interceptar el SCD.

D2-CAT5.2: Eavesdropping (ataque de canal indirecto). Los ataques de canal indirecto explotan el filtrado de información en base a las características del dispositivo hardware usado durante la ejecución de los algoritmos criptográficos. De esta manera, la clave privada puede llegar a conocerse. La importancia de este tipo de ataques radica en que la complejidad o robustez del algoritmo criptográfico no afecta al éxito del ataque, dado que el fundamento de los ataques de canal indirecto reside en la dependencia entre la información procesada (p. e. la clave privada de firma) y/o las operaciones realizadas por el dispositivo (p. e. tarjeta inteligente) con el comportamiento del hardware subyacente.

D2-CAT5.2.1: Tiempo. Un ataque de análisis de tiempo explota los tiempos de ejecución medidos para un cierto número de operaciones.

D2-CAT5.2.2: EMA. Un ataque de análisis electromagnético (EMA) explota la correlación entre los datos secretos y las variaciones en las radiaciones emitidas por los dispositivos.

D2-CAT5.2.3: Potencia. Un ataque de análisis de potencia analiza la relación entre el consumo de potencia de un dispositivo criptográfico y los datos manejados durante las operaciones criptográficas.

D2-CAT5.2.4: Microarquitectural. Un ataque de análisis microarquitectural estudia los efectos que los componentes pertenecientes a los procesadores comunes y su funcionalidad tienen sobre la seguridad de los criptosistemas software.

D2-CAT5.2.5: Observación óptica. Las emanaciones ópticas también pueden filtrar información al atacante. En caso que los datos procesados por un dispositivo que emanara este tipo de señales se correspondieran con el SCD, el atacante podría comprometerlo.

D2-CAT5.3: Acceso no autorizado al SCDev. El atacante compromete el SCD accediendo al SCDev donde se almacena.

D2-CAT5.3.1: Compromiso de datos de autenticación de firmante (SAD). El atacante es capaz de extraer el SCD una vez conocido el SAD. Este método de ataque requiere que el SCD sea exportable. Esta subcategoría se ramifica a su vez en las mismas subcategorías que *D2-CAT4.1 Compromiso de datos de autenticación de firmante (SAD)*.

D2-CAT5.3.2: Evitación de Autenticación. El atacante es capaz de acceder al SCD (y leerlo) incluso sin necesidad de conocer el SAD. Este método de ataque requiere que el SCD pueda ser leído.

D2-CAT5.4: Criptoanálisis. El atacante aplica métodos de criptoanálisis para conocer el SCD.

D2-CAT5.4.1: Algoritmo asimétrico. Esta subcategoría recopila cualquier ataque centrado en comprometer la clave privada usada en el algoritmo asimétrico concreto. Existen múltiples algoritmos asimétricos que pueden emplearse para realizar una firma electrónica basada en criptografía (p. e. RSA, DSA, curva elíptica, etc.). Dependiendo del algoritmo, el conjunto de posibles métodos de ataque variará.

No todos los métodos de ataque pueden permitir al atacante alcanzar los tres objetivos establecidos en la primera dimensión. El Cuadro I relaciona las categorías de la primera dimensión con las categorías de primer nivel de la segunda dimensión.

IV-C. Dimensión Patrón de Ataque

Un patrón de ataque describe cómo un tipo de ataque observado se lleva a cabo. Es la descripción del mecanismo empleado para explotar un sistema pero desde el punto de vista del atacante. CAPEC (Common Attack Pattern Enumeration and Classification) [10] es una extensa colección de patrones de ataque que se considera un referente a nivel mundial. Es una base de datos pública y cuyos patrones de ataque se generan tras un análisis exhaustivo de ataques reales. Para cada patrón de ataque se proporcionan datos relevantes como el flujo de ejecución del ataque o los requisitos que deben cumplirse para que el ataque pueda ser llevado a cabo.

En esta dimensión se incluye una selección de patrones de ataque de entre los 287 existentes actualmente en CAPEC. El criterio de selección no ha sido otro que el análisis de aplicabilidad de cada patrón de ataque en base al modelo de sistema planteado, y cuya instanciación pudiera comprometer la seguridad de un proceso de generación de firma. Se han identificado 192 patrones de ataque válidos que, por cuestiones de espacio, no se incluyen en el artículo.

IV-D. Dimensión Objeto del Ataque

Esta dimensión categoriza los posibles elementos que pueden ser objeto del ataque. Un ataque puede afectar a uno o varios elementos, por lo que la clasificación de un ataque podría necesitar la selección de varias categorías de esta dimensión (véase Sección IV-E).

Por cuestiones de espacio y relevancia en relación con otras dimensiones, no se proporciona un listado de las categorías existentes ni una descripción de las mismas. Simplemente se indican a continuación las categorías de primer nivel: D4-CAT1: Criptografía, D4-CAT2: Software, D4-CAT3: Hardware, D4-CAT4: Usuario.

IV-E. Método de Clasificación

El método de clasificación de una taxonomía debe guiar a un usuario de forma clara y unívoca en la clasificación de un nuevo elemento. Para clasificar un ataque en base a la taxonomía aquí presentada, deben seguirse los siguientes pasos:

1. Debe identificarse el objetivo del ataque, y clasificarse de acuerdo a la dimensión *Objetivo del Atacante*.
2. Una vez conocido el objetivo, debe clasificarse el método empleado por el atacante para alcanzar dicho objetivo, de acuerdo a la dimensión *Método de Ataque*, y la Tabla I. El método debe clasificarse en la subcategoría de mayor profundidad posible.
3. El patrón de ataque concreto empleado debe seleccionarse de acuerdo a la dimensión *Patrón de Ataque* y la información existente en la base de datos de CAPEC. Al igual que en el caso del método, debe seleccionarse el patrón de ataque más concreto posible.
4. Deben identificarse los elementos del SCE que se ven afectados de forma directa o indirecta por el ataque, y clasificarse en base a la dimensión *Objeto del Ataque*. En caso que el ataque afecte a más de un elemento, deberán seleccionarse varias instancias (categorías) de esta dimensión.
5. En cada paso, si se detecta la necesidad de añadir una nueva categoría o subcategoría, ésta debe incorporarse a la taxonomía, y clasificar el ataque como corresponda.

Así pues, un ataque se clasificará con una categoría de la dimensión *Objetivo del Atacante*, una (sub)categoría de la dimensión *Método de Ataque*, una (sub)categoría de la dimensión *Patrón de Ataque*, y una o varias (sub)categorías de la dimensión *Objeto del Ataque*.

V. Evaluación de la Taxonomía

Una taxonomía debe satisfacer una serie de requisitos generales [18]. Una taxonomía debería ser *ampliamente aceptada* en el campo de aplicación. La taxonomía propuesta se fundamenta en trabajos previos que han tenido un fuerte impacto en la comunidad científica. Así mismo, la taxonomía aplica el diseño basado en dimensiones, que ha demostrado ofrecer una perspectiva integral del campo de estudio abordado. Por ello, y por la falta actual de una taxonomía centrada en ataques a firma electrónica, creemos que la taxonomía aquí propuesta será un primer paso en el estudio de esta problemática.

La taxonomía debería ser *exhaustiva* en el sentido de que pudiera cubrir la clasificación de cualquier elemento dentro del campo de estudio. Este requisito es difícil de cumplir dado que no es posible conocer todos los ataques existentes a firmas electrónicas, especialmente en un campo tan dinámico como el de la seguridad. Sin embargo, la evaluación de la taxonomía frente ataques reales es fundamental para verificar su completitud. En este sentido, se ha llevado a cabo de forma satisfactoria una clasificación de 70 ataques encontrados en la literatura, pero que, por motivos de espacio, no se incluyen en el artículo.

Las categorías de la taxonomía deberían ser *excluyentes entre sí*, de forma que cada ataque fuera clasificado exclusivamente en una categoría de cada dimensión. El diseño de la taxonomía y el método de clasificación aseguran este principio. La posibilidad de seleccionar varias categorías en la dimensión *Objeto del Ataque* no implica que se viole este requisito, sino

Objetivo	Método
D1-CAT1: Firma no consciente de datos	D2-CAT1: Manipulación del entorno D2-CAT2: Modificación previa a la generación de la firma
D1-CAT2: Uso no autorizado de los datos de creación de firma (SCD)	D2-CAT4: Invocación no autorizada de la operación de firma D2-CAT5: Compromiso de datos de creación de firma (SCD)
D1-CAT3: Sustitución/Modificación de datos firmados	D2-CAT3: Modificación posterior a la generación de la firma

Cuadro I
RELACIÓN ENTRE DIMENSIÓN OBJETIVO Y DIMENSIÓN MÉTODO

que la taxonomía permite la clasificación de varios elementos afectados por el ataque si fuera necesario.

El método de clasificación debería ser *determinista* y *repetible* en base al diseño de la taxonomía. El método aquí proporcionado es claro, sencillo e inequívoco teniendo en cuenta las dimensiones diseñadas.

La taxonomía debería emplear *terminología ampliamente aceptada* y ser *apropiada*, basándose en un modelo claro de referencia. La terminología y modelo de sistema empleados en la presente taxonomía se derivan de los estándares actuales. Así mismo, el modelo acota exactamente los elementos de alto nivel que intervienen y las relaciones entre ellos.

Para ser útil, una taxonomía debería estar *especializada* en un área de conocimiento concreta. En nuestro caso, la taxonomía se centra en ataques a entornos de creación de firmas electrónicas basadas en criptografía de clave pública.

Por último, la taxonomía debería ser *útil*. En este sentido, la taxonomía propuesta permite la clasificación sistemática de ataques a entornos de creación de firmas, lo cual sin duda redundará en un mayor conocimiento a la hora de diseñar soluciones más eficaces no sólo contra ataques conocidos sino contra ataques potenciales todavía por aparecer.

VI. CONCLUSIONES

La importancia de la firma electrónica, con su amplio reconocimiento por las legislaciones y estándares internacionales vigentes, así como las consecuencias que de ella se derivan, nos llevan a la necesidad de diseñar sistemas y soluciones robustos ante las amenazas existentes y futuras, por otra parte numerosas. Una taxonomía de ataques permitiría categorizar los ataques posibles de forma genérica y abstracta, beneficiando el estudio de contramedidas globales aplicables a cualquier entorno de firma.

Aunque se han propuesto numerosas taxonomías en el ámbito de la seguridad, ninguna ha abordado de manera integral el problema de seguridad de las firmas electrónicas. En este artículo se ha presentado la primera taxonomía de ataques a entornos de creación de firma electrónica. La taxonomía se ha dividido en cuatro dimensiones, lo cual permite el análisis y clasificación de ataques desde un punto de vista holístico. La taxonomía se ha validado satisfactoriamente frente a los requisitos y principios fundamentales para taxonomías, incluyendo el requisito de completitud mediante la clasificación de setenta ataques publicados en la literatura (no incluidos en el presente artículo).

Como trabajo futuro se ampliará la taxonomía incorporando categorías de ataque orientados a la fase de verificación (p. e. aprovechar la latencia en la actualización de CRL para firmar con un certificado revocado). Como resultado, se dispondrá de una taxonomía completa de ataques a la firma electrónica, posibilitando la clasificación y estudio de cualquier ataque que afecte a su fiabilidad como evidencia de no repudio.

REFERENCIAS

[1] A. McCullagh and W. Caelli. Non-repudiation in the digital Environment. First Monday, vol. 5, no. 8, 2000.

[2] European Directive 1999/93/CE of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures, 1999.

[3] Ley 59/2003, de 19 de Diciembre, de Firma Electrónica, 2003.

[4] UNCITRAL Model Law on Electronic Signatures with Guide to Enactment, United Nations, 2001.

[5] Electronic Signatures in Global and National Commerce Act (e-sign). Federal Trade Commission, Department of Commerce. United States of America. 30 Junio, 2000.

[6] Personal Information Protection and Electronic Documents Act. Department of Justice. Government of Canada. 30 Mayo, 2008.

[7] D. Cruz Rivero. Eficacia formal y probatoria de la firma electrónica. Marcial Pons (Ed.). ISBN: 84–9768–353–6, 2006.

[8] ISO/IEC 13888-3 Information technology – Security techniques – Non repudiation – Part 3: Mechanisms Using Asymmetric Techniques. 2009.

[9] CEN CWA 14170 – Security Requirements for signature creation applications. The European Committee for Standardization (CEN), 2004.

[10] Common Attack Pattern Enumeration and Classification (CAPEC) - A Community Knowledge Resource for Building Secure Software. MITRE.

[11] A. Axelsson. Intrusion Detection Systems: a Survey and Taxonomy. Technical Report No 99-15, Department of Computer Engineering, Chalmers University, Gothenburg, 2000.

[12] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi. A taxonomy of computer program security flaws. ACM Computing Surveys, vol. 26, no. 3, pp. 211–254, 1994.

[13] M. Bishop. A taxonomy of (Unix) system and network vulnerabilities. Technical Report CSE–9510. Department of Computer Science, University of California, 1995.

[14] F. Piessens. A taxonomy of causes of software vulnerabilities in Internet software. 13th International Symposium on Software Reliability Engineering, pp. 47–52, 2002.

[15] U. Lindqvist and E. Johsson. How to Systematically Classify Computer Security Intrusions. IEEE Security and Privacy, pp. 154–163. 1997.

[16] H. Langweg and E. Snekkenes. A Classification of Malicious Software Attacks. Proceedings of 23rd IEEE International Performance, Computing, and Communications Conference, 2004.

[17] S. Hansman. A Taxonomy of Network and Computer Attack Methodologies, technical report, Department of Computer Science and Software Engineering, University of Canterbury, Christchurch, 2003.

[18] D. L. Lough. A taxonomy of computer attacks with applications to wireless networks. Tesis Doctoral, 2001.

[19] K. Kain. Electronic Documents and Digital Signatures. Doctoral Thesis, 2003.

[20] P. Girard, J-L. Giraud. Software attacks on smart cards. Information Security Technical Report, vol. 8, no. 1, pp. 55–66. 2003.

[21] A. J. Rae, L. P. Wildman. A Taxonomy of attacks on secure devices. Proceedings of the Fourth Australian Information Warfare and IT Security Conference, pp. 251–264. 2003.

[22] The side-channel cryptanalysis lounge. European Network of Excellence in Cryptology.

Envío de información con soporte de firma digital y cifrado desde un dispositivo móvil a un servidor web

Jan Bühler Olivé Macià Mut Puigserver Magdalena Payeras Capellà Llorenç Huguet Rotger
Universitat de les Illes Balears Universitat de les Illes Balears Universitat de les Illes Balears Universitat de les Illes Balears
Email: jan.buhler@uib.es Email: macia.mut@uib.es Email: mpayeras@uib.es Email: l.huguet@uib.es

***Abstract*—En este artículo se presenta el diseño y la implementación de una solución para el envío de información con soporte de firma digital y cifrado desde un dispositivo móvil a un servidor web. El artículo no sólo introduce los fundamentos técnicos de las herramientas y tecnologías utilizadas, sinó que también presenta el estado del arte de este tipo de tecnologías y las características, ventajas y desventajas de otras soluciones y en especial de la solución presentada en este trabajo de investigación. Finalmente, se analiza el protocolo implementado haciendo especial referencia a las propiedades de seguridad introducidas en él y a las técnicas aplicadas para conseguir este fin.**

I. Introducción

El acceso a Internet a través de dispositivos móviles es cada vez más utilizado. Distintos estudios indican que en los próximos años casi la mitad de los usuarios se conectarán a Internet a través de dispositivos móviles.

El sector de las telecomunicaciones se encamina progresivamente a la convergencia de las redes fija y móvil. Los fabricantes de teléfonos están desarrollando terminales con acceso a Internet, correo electrónico, etc. y las operadoras lanzan al mercado tarifas de telefonía móvil e Internet a precios cada vez más competitivos.

Actualmente ya existen multitud de dispositivos móviles que disponen de una interfaz de acceso Wi-Fi con la que es posible acceder a Internet a un precio más económico. El único inconveniente es la necesidad de un punto de acceso, por lo que se deja de ser completamente *móvil*.

En un futuro no muy lejano se espera disponer de una conexión móvil para acceder a Internet, es decir, se podrá acceder desde el lugar que sea, desde el dispositivo que sea y en el momento que sea preciso.

Todos estos avances provocan la necesidad y la demanda de aplicaciones y servicios en terminales móviles como por ejemplo: consultar el correo electrónico, consultar notícias, visualizar mapas y callejeros, enviar imágenes y vídeos, etc.

En este artículo presentamos el diseño de una aplicación para plataformas móviles, concretamente una aplicación para el envío de texto e imágenes desde un dispositivo móvil a un servidor web pero con el valor añadido de que los mensajes son enviados cifrados y firmados digitalmente. Con este fin hemos definido un protocolo de comunicación entre terminal y servidor donde la seguridad juega un papel muy importante.

En realidad se han desarrollado dos aplicaciones, una para el emisor (dispositivo móvil) y una para el receptor (servidor web). Para el desarrollo de la aplicación del emisor se ha utilizado la novedosa plataforma de programación de software para dispositivos móviles Android desarrollada primero por Google y luego por la *Open Handset Alliance* [9], mientras que para el receptor se ha utilizado el servidor web Tomcat con soporte a Java servlets y JSPs [3]. La gran ventaja de esta configuración es poder utilizar el mismo lenguaje de programación (Java) en ambas aplicaciones (emisor y receptor), lo cuál facilita enormemente el desarrollo y el correcto funcionamiento de las operaciones de cifrado y descifrado de datos y de firma y verificación de firma digital.

A. Seguridad

Uno de los principales objetivos del proyecto ha sido construir un protocolo que introdujera seguridad en la comunicación emisor-receptor. En el extremo del emisor, el protocolo consiste en: construir el mensaje a enviar, firmarlo digitalmente utilizando técnicas de criptografía asimétrica y funciones resumen (hash), cifrarlo utilizando técnicas de criptografía simétrica, comprimirlo para reducir el tráfico de red y enviarlo. En cuanto al extremo del receptor, el protocolo es el siguiente: descomprimir el mensaje recibido, descifrarlo utilizando técnicas de criptografía simétrica, verificar la validez de los certificados digitales utilizados y verificar la firma digital mediante técnicas de criptografía asimétrica y funciones resumen, enviar una evidencia de recepción del mensaje al emisor a través de correo electrónico y tratar el mensaje. Mediante esta implementación se ha dotado al protocolo de las propiedades de integridad, confidencialidad, autenticidad, no repudio de origen y no repudio de recepción. Para lograrlo, también se ha desarrollado una infraestructura de clave pública (PKI) ([12], [13]), es decir, un sistema para la gestión de certificados digitales y aplicaciones de firma digital.

B. Aplicación del terminal móvil

Es importante tener en cuenta que el desarrollo de aplicaciones para dispositivos móviles resulta más complicado que el desarrollo tradicional de aplicaciones web o de escritorio ya que, está limitado tanto en la capacidad de recursos hardware (principalmente de memoria y de procesador) como en el

uso de librerías y funcionalidades propias del lenguaje de programación para dispositivos móviles utilizado.

Otro aspecto importante del trabajo que presentamos aquí ha sido el relativo al rendimiento de la aplicación del emisor (dispositivo móvil), especialmente por la carga que suponen las operaciones criptográficas. Como durante todo el desarrollo del proyecto se ha trabajado sobre un emulador, se ha intentado dar respuesta a la pregunta ¿soportaría un terminal real, con sus limitaciones hardware, la carga generada por estas operaciones criptográficas?. La conclusión de la investigación llevada a cabo es que no es posible juzgar el rendimiento de la aplicación en base al emulador, ya que a pesar de poder hacer estimaciones en referencia a la CPU, no existe emulación del bus de memoria, no existe aceleración gráfica por hardware, etc. Por tanto, la única manera de conocer el comportamiento de la aplicación realmente es ejecutándola sobre un dispositivo real. Apuntar que, de forma definitiva, la aplicación desarrollada se ha trasladado del emulador al terminal físico y el rendimiento ha resultado ser satisfactorio, teniendo en cuenta la carga operacional requerida.

Finalmente, decir que modificando mínimamente la aplicación del emisor y adaptando la aplicación del receptor para realizar el correspondiente tratamiento de los mensajes recibidos, este proyecto es válido para numerosas aplicaciones.

II. Entorno de desarrollo

En esta sección se describe el entorno de desarrollo y las distintas tecnologías utilizadas en este proyecto.

Tanto para el desarrollo de la aplicación emisor como de la aplicación receptor se ha utilizado:

- El sistema operativo GNU-Linux (*release 2.6.24-23-generic*), concretamente la distribución Ubuntu 8.04 - *Hardy Heron* - publicada en abril de 2008.
- La plataforma de desarrollo: Java Standard Edition 6 JDK (*product version 6, development version 1.6.0*).
- El entorno de desarrollo integrado Eclipse que, al estar formado por módulos (*plugins*), ofrece una plataforma ligera para componentes de software. De este modo, cada usuario puede instalar y activar únicamente los plugins que precisa. Para el desarrollo de la aplicación emisor se ha utilizado Eclipse junto al plugin ADT (*Android Development Tools*) que incluye una serie de extensiones para facilitar la creación, ejecución y depuración de aplicaciones Android, mientras que en la aplicación receptor, se ha utilizado Eclipse para la implementación de los Java servlets de recepción y publicación, y las correspondientes clases satélite.

Para el desarrollo de la aplicación emisor se ha instalado el kit de desarrollo Android SDK (*Android 1.0 SDK, Release 1*) proporcionado por Google. En este kit se encuentran todas las herramientas necesarias (entorno de depuración, librerías, emulador de terminal móvil, documentación, tutoriales, código de ejemplo, etc.) para poder desarrollar aplicaciones para la plataforma Android [11]. Android es una solución completa de software de código libre para teléfonos y dispositivos móviles constituida básicamente por un sistema operativo, un conjunto de librerías de bajo nivel como SQLite para la persistencia de datos, OpenGL ES para la gestión de gráficos 3D, Webkit como motor de rendeado HTML, etc., y un conjunto inicial de aplicaciones orientadas al usuario final. Las aplicaciones Android están programadas en Java pero no corren sobre J2ME, sino sobre Dalvik [7], una máquina virtual Java optimizada para dispositivos empotrados desarrollada expresamente por Google.

Por otro lado, comentar que la aplicación receptor se ejecuta sobre un servidor web, concretamente se ha instalado el servidor web con soporte de servlets y JSPs Apache Tomcat (*versión 6.0.18*).

A continuación se describen brevemente las herramientas, APIs y algoritmos utilizados en la implementación de las aplicaciones emisor y receptor:

- **JDOM** [4]**:** API que se ha utilizado para el acceso, la manipulación y la salida de datos XML.
- **Clase Base64:** clase Java que se ha utilizado para la codificación y decodificación de información en base 64. Antes de poder embeber una imagen, una firma digital o un contenido cifrado en un fichero XML debe codificarse en base 64.
- **RSA** [1]**:** sistema criptográfico de clave pública que se ha utilizado para implementar la *envoltura digital*[1] y las operaciones de firma digital.
- **AES** [14]**:** esquema de cifrado por bloques utilizado para el cifrado de los datos. Concretamente, se ha utilizado en modo de cadena de cifrado de bloque (CBC).
- ***java.util.zip*:** paquete que permite la compresión y descompresión de ficheros. Para paliar la expansión producida por las distintas codificaciones en base 64, se decidió comprimir el fichero a enviar.

Finalmente, para la gestión de los diferentes *keystores*[2], de los pares de claves y los certificados digitales se han utilizado las herramientas Keytool IUI [5] y OpenSSL [10].

III. Estado del arte

Como veremos a continuación, actualmente ya existen diversas soluciones de firma electrónica y cifrado sobre dispositivos móviles. La firma electrónica y el cifrado móvil pueden aportar autenticidad, integridad, no repudio y confidencialidad si se utilizan en el entorno adecuado de un protocolo de comunicación.

En la telefonía comercial se ofrece un servicio de firma electrónica móvil ([8], [17]), que permite autenticar al usuario y firmar transacciones y documentos digitales de todo tipo mediante el móvil. Las desventajas de este servicio son su elevado coste y que de momento sólo se ofrece a clientes empresa.

[1]Envoltura digital: técnica utilizada en la negociación de claves simétricas mediante criptografía asimétrica. Consiste en utilizar un método seguro para compartir claves (por ejemplo RSA) y un método rápido para cifrar los datos (por ejemplo AES).

[2]Keystore: un almacén de claves es un archivo de base de datos de claves que contiene tanto las claves públicas como las privadas.

Por otro lado, la empresa tailandesa CryptoGraf [2] ofrece un producto que permite el envío y la recepción de mensajes SMS y MMS cifrados y firmados digitalmente entre usuarios de la aplicación. La desventaja de este producto es el aumento de tamaño de los mensajes (*overhead*) debido al cifrado y a la firma digital, que provoca que éstos deban fraccionarse y enviarse por partes. Por ejemplo, en el caso de enviar un mensaje SMS cifrado y firmado digitalmente, el *overhead* es de como mínimo tres mensajes.

Otra empresa, en este caso la española Lleida.net [6] ofrece un servicio de notificaciones certificadas vía SMS que se utiliza para notificar multas o sanciones como sustituto de las cartas certificadas o los burofaxes. Las principales ventajas de este servicio son su bajo coste, la rapidez del proceso de certificación y el ahorro de papel y tinta. La desventaja es que de momento sólo permite el envío de SMSs.

En este proyecto se han mantenido las ventajas de las soluciones existentes y se han eliminado sus desventajas. Una de las principales ventajas de la solución presentada aquí es su versatilidad, ya que realizando una serie de pequeñas modificaciones pueden implementarse diversos escenarios. Como la conexión entre emisor y recepetor es a través de internet, puede enviarse prácticamente cualquier tipo de contenido digital. Por ejemplo: el usuario se podría descargar un documento, firmarlo digitalmente y enviarlo al receptor. Tampoco resultaría complicado incluir una autoridad selladora de tiempo para tener constancia ante terceros del momento en que se realiza la firma digital.

La implementación final de este proyecto se ha orientado a la publicación móvil segura de imágenes. Existen diversos escenarios donde podría resultar útil, como por ejemplo en la gestión de las infracciones de tráfico o de aparcamiento. Los agentes al detectar una infracción, como por ejemplo un coche mal aparcado, procederían a tomar una fotografía de la misma y la enviarían al receptor junto a una serie de datos, como su número de agente, el número de matrícula del coche implicado, y la descripción de la infracción.

Actualmente, en la web 2.0, existen numerosas soluciones de publicación de imágenes a través de dispositivos móviles pero no con soporte de cifrado y firma digital como se ofrece en este proyecto. En algunos casos la publicación de imágenes se lleva a cabo enviando la imagen por correo electrónico a una dirección única de carga. En el trabajo que hemos desarrollado aquí se realiza mediante una conexión directa con el receptor, utilizando el método POST del protocolo HTTP.

IV. Solución implementada

Con el fin de realizar una descripción de la forma más clara y concreta posible, en esta sección se describe la solución diseñada como caso específico de implementación del protocolo genérico de comunicaciones seguras entre terminal móvil y servidor desarrollado.

A. Nuestras playas online

En este caso concreto hemos desarrollado una aplicación que hemos llamado *Nuestras playas online*. En esta aplicación los emisores, que pueden ser los propios vigilantes de la playa, van enviando periódicamente fotos de la playa en la que se encuentran junto a su información de estado, concretamente la bandera, la temperatura, el viento, si hay medusas u otros animales que podrían ser considerados peligrosos para los bañistas, el oleaje, si hay corriente, si hay posidonia y la cantidad de gente.

El receptor (que aquí sería el servidor de la autoridad competente en costas), al recibir un mensaje lo trata y publica la información correspondiente en una página web, de modo que cualquier ciudadano pueda consultar el estado de las playas antes de salir de casa.

B. Protocolo de transmisión

1) Notación: En la tabla 1 especificamos el significado de los elementos utilizados durante la emisión y recepción de la información. En esta tabla, el símbolo '+' se utiliza para indicar la concatenación de los elementos implicados.

`M`	Mensaje XML que contiene la imagen y la información que se pretende enviar
`AC`	Almacén de claves
PR_E	Clave privada del emisor
PU_E	Clave pública del emisor
PR_R	Clave privada del receptor
PU_R	Clave pública del receptor
`F`	Firma digital
F_a	Fecha actual
`H(M)`	Hash (resumen digital) del mensaje `M`
`MF`	Mensaje `M` firmado: `M + F` + F_a
`IV`	Vector de inicialización para el modo CBC del algoritmo AES
`SK`	Clave secreta para utilizar en el algoritmo AES
`CX(IV,SK,MF)`	Cifrar el mensaje `MF` utilizando el vector de inicialización `IV` y la clave secreta `SK`
PU_R`(SK)`	Cifrar la clave secreta `SK` mediante la clave pública del receptor PU_R
`MX`	Mensaje `M` firmado y cifrado: `IV + CX(IV,SK,MF) +` PU_R`(SK)`
`C(M)`	Compresión del mensaje `M`
`MC`	Mensaje `M` firmado, cifrado y comprimido
`D(MC)`	Descompresión del mensaje comprimido `MC`
PR_R`(MX)`	Recuperar la clave secreta `SK` del mensaje firmado y cifrado `MX` mediante la clave privada del receptor PR_R
`DX(IV,SK,MX)`	Descifrar el mensaje firmado y cifrado `MX` utilizando el vector de inicialización `IV` y la clave secreta `SK`
`FE(MF)`	Recuperar fecha de firma digital del mensaje firmado `MF`
C_E	Certificado de clave pública del emisor
`V(C, M)`	Verificar si el certificado `C` era válido al firmar el mensaje `M`
`V(`C_1`,` C_2`)`	Verificar si el certificado C_1 era válido al firmar el certificado C_2
`VF(PU, F, M)`	Verificación de la firma digital
`ENV(NRR, M)`	Enviar no repudio de recepción del mensaje `M`

Tabla 1. Descripción de los elementos implicados en el protocolo.

1. `M`	Crear mensaje
2. PR_E ⇐ `AC`	Recuperar clave privada del emisor
3. `F` ⇐ PR_E`(H(M))`	Firma digital del mensaje
4. `MF` ⇐ `M + F +` F_a	Construir mensaje firmado
5. `SK`	Generar clave secreta
6. `IV`	Generar vector de inicialización
7. PU_R ⇐ `AC`	Recuperar clave pública del receptor
8. `MX` ⇐ `IV + CX(IV,SK,MF) +` PU_R`(SK)`	Cifrar mensaje firmado
9. `MC` ⇐ `C(MX)`	Comprimir mensaje cifrado
10. `MC`	Enviar mensaje comprimido

Tabla 2. Pasos del protocolo de transmisión realizados por la aplicación emisor.

2) Descripción:

Emisión: En la tabla 2 enumeramos y describimos cada una de las operaciones que realiza la aplicación instalada en el dispositivo de comunicación móvil. El objetivo principal de la aplicación emisor es enviar una imagen junto a unos determinados datos a la aplicación receptor, con la peculiaridad de que la información enviada está firmada digitalmente, cifrada y comprimida.

A continuación se describen detalladamente los pasos del protocolo realizados en la aplicación emisor:

- **Paso 1:** Construcción del fichero XML a enviar (M):
  - Seleccionar la imagen a publicar.
  - Embeber la imagen codificada en base 64 en un fichero XML temporal: *tmp.xml* (M).
  - Añadir el alias del usuario, la dirección de correo electrónico donde recibir la evidencia de recepción y, en el caso concreto de la aplicación desarrollada, se añaden al fichero *tmp.xml* (M) el resto de los datos, como por ejemplo: la isla, la playa, la bandera, etc.
- **Pasos 2, 3 y 4:** Firmar el fichero *tmp.xml* (M) construido, obteniendo el fichero *tmpFirmado.xml* (MF):
  - Recuperar la clave privada del emisor (PR_E) del almacén de claves (AC).
  - Firmar el fichero *tmp.xml* construido utilizando la clave privada del emisor (PR_E).
  - Añadir la firma en base 64 al contenido del fichero *tmp.xml* (M).
  - Añadir la fecha actual (F_a) al contenido del fichero *tmp.xml* (M) para dejar constancia de cuándo ha sido firmado.
  - Almacenar el fichero resultante como *tmpFirmado.xml* (MF).
  - Si ocurre un error durante este proceso (por ejemplo si la contraseña para acceder al almacén de claves es incorrecta) se construye un documento XML de error con el alias, el error y la fecha actual (M ⇐ alias + error + fecha actual), que es enviado en lugar del fichero XML construido anteriormente (se denomina *tmpFirmado.error*). De este modo el receptor puede detectar si se producen intentos fallidos de acceso a la clave privada de un determinado emisor.
- **Pasos 5, 6, 7 y 8:** Cifrar el fichero *tmpFirmado.xml* (MF) obteniendo el fichero *tmpCifrado.xml* (MX):
  - Generar clave secreta (SK) de un sólo uso de 256 bits (*Unlimited Strength Cryptography* [16]).
  - Generar el vector de inicialización (IV), para poder utilizar el modo CBC del algoritmo AES.
  - Cifrar el fichero *tmpFirmado.xml* (MF) utilizando el algoritmo AES con la clave secreta (SK) y el vector de inicialización (IV) generados.
  - Crear el fichero *tmpCifrado.xml* (MX) con el vector de inicialización (IV) concatenado al contenido del fichero *tmpFirmado.xml* (MF) cifrado.
  - Recuperar la clave pública del receptor (PU_R) de su certificado digital que se encuentra almacenado en el almacén de claves (AC).
  - Cifrar la clave secreta (SK) con la clave pública del receptor (PU_R), de forma que sólo éste podrá descifrarla mediante su clave privada.
  - Añadir la clave secreta (SK) cifrada con la clave pública del receptor (PU_R) al fichero *tmpCifrado.xml* (MX).
- **Paso 9:** Comprimir el fichero *tmpCifrado.xml* (MX) para optimizar el envío y paliar el incremento de tamaño producido por las distintas codificaciones en base 64 realizadas, obteniendo el fichero *tmpCifrado.gzip* (MC).
- **Paso 10:** Enviar el fichero *tmpCifrado.gzip* (MC) estableciendo una conexión con el servlet de recepción de la aplicación receptor utilizando el método POST del protocolo HTTP.

Recepción: En la tabla 3 expresamos de forma esquemática cada una de las operaciones que realiza el servidor en la recepción de los mensajes.

1. MC	Recepción del mensaje comprimido
2. MX ⇐ D(MC)	Descomprimir, obteniendo el mensaje cifrado
3. IV ⇐ MX	Recuperar vector de inicialización
4. PR_R ⇐ AC	Recuperar clave privada del receptor
5. SK ⇐ PR_R(MX)	Recuperar clave secreta
6. MF ⇐ DX(IV,SK,MX)	Descifrar, obteniendo mensaje firmado
7. (F_a − T_m) < FE(MF) < F_a?	Verificar validez del periodo margen
8. C_E ⇐ AC	Recuperar certificado digital del emisor
9. V(C_E,MF)?	Verificar validez del certificado digital del emisor en el momento de firmar el mensaje
10. C_{CA} ⇐ AC	Recuperar certificado digital de la autoridad certificadora
11. V(C_{CA},C_E)?	Verificar validez del certificado digital de la autoridad certificadora en el momento de firmar el certificado de usuario
12. PU_E ⇐ AC	Recuperar la clave pública del emisor
13. F ⇐ MF	Recuperar la firma digital del mensaje firmado
14. M ⇐ MF	Recuperar el mensaje original del mensaje firmado
15. VF(PU_E,F,M)?	Verificar la firma digital
16. ENV(NRR. M)	Enviar evidencia de recepción
17. M	Tratamiento del mensaje

Tabla 3. Pasos del protocolo de transmisión realizados por la aplicación receptor.

La aplicación receptor realiza dos funciones bien diferenciadas:

- Recibir y publicar los mensajes enviados desde las aplicaciones emisoras (dispositivos móviles) a través de una interfaz HTTP POST implementada mediante un servlet.
- Proporcionar acceso vía web a la información publicada a través de un conjunto de páginas HTML y un servlet que carga la información desde el fichero XML correspondiente.

A continuación se describen detalladamente los pasos del protocolo realizados en la aplicación receptor:

- **Paso 1:** Recepción del fichero *tmpCifrado.gzip* (MC). Como pueden llegar múltiples archivos simultáneamente es necesario identificarlos unívocamente por lo que el fichero recibido es renombrado como *idUnico.gzip*.

- **Paso 2:** Descomprimir el fichero *idUnico.gzip* (MC) obteniendo el fichero *idUnicoCifrado.xml* (MX).
- **Pasos 3, 4, 5 y 6:** Descifrado del fichero *idUnicoCifrado.xml* (MX):
  - Recuperar el vector de inicialización (IV) del fichero *idUnicoCifrado.xml*.
  - Recuperar la clave privada del receptor (PR_R) del almacén de claves (AC).
  - Extraer la clave secreta (SK) del fichero *idUnicoCifrado.xml* utilizando la clave privada del receptor (PR_R) (sólo éste puede decifrarla).
  - Descifrar el contenido del fichero *idUnicoCifrado.xml* mediante el vector de inicialización (IV) y la correspondiente clave secreta (SK), obteniendo el fichero *idUnicoFirmado.xml* (MF).
- **Paso 7**[3]**:** Comprobar si la fecha de firma no es anterior al periodo margen (T_m) ni posterior a la fecha actual (F_a):
  - Utilizar la fecha de firma incluida en el fichero *idUnicoFirmado.xml* (MF) y la fecha actual (F_a).
- **Pasos 8 y 9:** Comprobar la validez del certificado de usuario en el momento de realizar la firma digital:
  - Recuperar el certificado del usuario emisor (C_E) del almacén de claves (AC).
  - Comprobar si el certificado era válido en el momento de firmar el fichero *idUnicoFirmado.xml* (MF) (éste contiene la fecha de firma).
- **Pasos 10 y 11:** Comprobar la validez del certificado de la autoridad certificadora en el momento de firmar el certificado de usuario:
  - Recuperar el certificado del usuario emisor (C_E) del almacén de claves (AC).
  - Recuperar la fecha de creación del certificado de usuario del emisor (C_E).
  - Recuperar el certificado de la autoridad certificadora (C_A).
  - Comprobar si el certificado de la autoridad certificadora (C_A) era válido en el momento de firmar el certificado de usuario del emisor (C_E).
- **Pasos 12, 13, 14 y 15:** Verificar la firma digital del fichero *idUnicoFirmado.xml* (MF):
  - Recuperar la clave pública del emisor (PU_E) del almacén de claves (AC).
  - Recuperar la firma del fichero (F) *idUnicoFirmado.xml*.
  - Recuperar el texto que fue firmado (M) del fichero *idUnicoFirmado.xml*.
  - Verificar la firma digital.
  - Si la verificación de firma es correcta, crear el fichero *idUnico.xml* (M) que contiene únicamente el contenido que el emisor pretendía enviar al receptor.
  - Si la verificación de firma es incorrecta se registra en el fichero de *logging*.
- **Pasos 16 y 17:** Si la verificación de la firma digital ha sido correcta, tratar la información recibida contenida en el fichero *idUnico.xml* (M) y enviar al emisor la correspondiente evidencia de recepción (NRR) del mensaje:
  - Extraer la imagen del fichero *idUnico.xml* y almacenarla en la ubicación correspondiente.
  - Extraer el alias, la dirección de correo electrónico, la isla, la playa, la bandera, etc. del fichero *idUnico.xml* (M) y realizar las correspondientes tareas de publicación.
  - Enviar a la dirección de correo electrónico solicitada por el emisor una evidencia de recepción (NRR) (no repudio de recepción).

[3]Según el tipo de aplicación final, podría ser interesante disponer de un servicio de sellado temporal, es decir, en lugar de enviar directamente el fichero al receptor se enviaría antes a una autoridad de sellado de tiempo que incluiría un sello temporal en el fichero en cuestión. De este modo se tendría constancia ante terceros del momento en que se realizó la firma digital. En la implementación actual lo que se ha hecho es incluir la fecha en que se realizó la firma digital en el fichero enviado y en el receptor se verifica que como máximo la firma tenga una antigüedad de T_{margen} y no sea posterior a la fecha actual, evitando así la posibilidad de recibir una fecha falsa, bastante anterior a la real o posterior a la fecha actual, y eludir por tanto la manipulación de la comprobación de la validez de los certificados.

V. Análisis informal de la seguridad del protocolo

Para que una transacción online se considere segura debe cumplir las propiedades de integridad, confidencialidad, no repudio en origen, no repudio de recepción y autenticación. En este proyecto se han utilizado técnicas de criptografía simétrica para el cifrado de los mensajes enviados del emisor al receptor. Concretamente, se ha utilizado el esquema de cifrado por bloques AES con una clave secreta de 256 bits. La criptografía simétrica dota al protocolo de la propiedad de condidencialidad. Para la implementación de la firma digital y el cifrado de la clave secreta, se han utilizado técnicas de criptografía asimétrica. Concretamente el sistema criptográfico de clave pública RSA con claves de 1024 bits. La criptografía asimétrica mediante la implementación de la firma digital dota al protocolo de las propiedades de integridad, confidencialidad y no repudio en origen. En el caso del dipositivo móvil, el par de claves es generado externamente por un operador de registro al dar de alta a un nuevo usuario y es almacenado en un almacén de claves de tipo BKS [15] junto al conjunto de ficheros que conforman la aplicación Android. En Android, por defecto, todos los datos asociados a una aplicación son privados para esa aplicación por lo que, en principio, la única forma de acceder al fichero que implementa el almacén de claves para intentar comprometer la seguridad del sistema sería accediendo al terminal como usuario *root*.

Finalmente, se ha implementado un mecanismo para dotar al protocolo de la propiedad de no repudio de recepción. Este mecanismo consiste en el envío (mediante correo electrónico) por parte del receptor al emisor de una notificación de recepción del mensaje. Por tanto, también queda constancia de la fecha de recepción del mensaje.

VI. Rendimiento del dispositivo móvil real

Para analizar el rendimiento del dipositivo móvil real se han realizado distintas ejecuciones de la aplicación implementada

capturando el tiempo de ejecución de las distintas operaciones realizadas (componer el mensaje, firmar digitalmente, cifrar, comprimir y enviar). Los resultados obtenidos se pueden observar en la tabla 4.

Tamaño ficheros (KB) - Tiempo Operaciones (ms)										
Imagen	Crear M	M	Firmar	MF	Cifrar	MX	Comp.	MC	Enviar	Total
99,3	2329	134,6	6277	134,9	5068	182,5	337	140,7	1151	15162
199,8	4894	270,4	9754	270,7	9373	366	798	282,3	867	25686
270,1	6409	365,3	11715	365,6	12295	494,2	881	381	1510	32810
368,1	8788	497,7	14352	498	16834	673,1	1247	519	1464	42685
465,5	10507	629,3	18642	629,6	20463	850,8	1839	656	1639	53090
517,6	13074	699,6	20072	699,9	23493	945,8	1627	729,3	2260	60526

Tabla 4. Tiempo consumido por dispositivo móvil real para realizar las distintas operaciones sobre ficheros de distintos tamaños.

Como se puede observar, las operaciones criptográficas consumen la mayor parte del tiempo, alrededor del 70% del mismo. La operación de composición del mensaje consume aproximadamente una quinta parte del tiempo total, debido esencialmente a la codificación en base 64 de la imagen, mientras que las operaciones de compresión y de envío consumen una parte muy reducida del tiempo.

Dado que los algoritmos de compresión se basan en la eliminación de la redundancia de datos se ha verificado que resulta más eficiente realizar antes la operación de compresión que la de cifrado, hecho que se tendrá en cuenta en futuras versiones del protocolo.

VII. Conclusiones y trabajos futuros

En este artículo se presenta una posible solución al problema de enviar información de forma segura desde un dispositivo móvil a un servidor web. Para lograrlo se han tenido que tomar una serie de decisiones y ha sido necesario el dominio de numerosas tecnologías, algoritmos criptográficos y herramientas distintas.

Durante el proceso de investigación se ha experimentado con numerosas plataformas de desarrollo para dispositivos móviles pero tarde o temprano siempre aparecía alguna limitación grave, como por ejemplo la imposibilidad de realizar todas las operaciones criptográficas requeridas en esta implementación, la imposibilidad de manipular ficheros XML, etc. Finalmente la plataforma Android ha permitido la implementación con las características deseadas.

Por otro lado, actualmente las principales barreras con las que se encuentran las soluciones como la presentada son el elevado coste de las comunicaciones móviles y los problemas de rendimiento de los terminales, pero parece ser que en un futuro no muy lejano dejarán de serlo. Actualmente, una solución para reducir el coste de la conexión a Internet consiste en conectarse a través de la interfaz Wi-Fi del terminal.

Entre las posibles mejoras que preveemos en el protocolo y la aplicación desarrollada se encuentran los siguientes trabajos futuros:

- Mejorar la infraestructura de clave pública (PKI), de forma que el usuario pueda generar su par de claves desde la aplicación emisor y enviar la correspondiente solicitud de firma del certificado.
- Modificar el protocolo de transmisión para añadir la propiedad de equidad a las propiedades de seguridad ya existentes.
- Dado que se ha verificado que resulta más eficiente realizar primero la operación de compresión que la de cifrado, intercambiar el orden de estas operaciones en el protocolo.
- Determinar el tamaño de las claves simétricas y asimétricas que ofrezcan la relación rendimiento-nivel de seguridad más adecuada.
- Ubicar el almacén de claves de la aplicación emisor en la tarjeta SIM para aumentar el nivel de seguridad y reducir el tiempo de cálculo.

Agradecimientos

Este trabajo ha sido parcialmente financiado por los proyectos ARES ”grupo de investigación avanzada en seguridad y privacidad de la información” (Consolider-Ingenio CSD 2007-00004) y ”Seguridad en la contratación electrónica basada en servicios web” (CICYT TSI 2007 62986).

Referencias

[1] R. Rivest, A. Shamir, L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, Vol. 21 (2), pp.120–126. 1978. Previously released as an MIT ”Technical Memo” in April 1977. Initial publication of the RSA scheme

[2] CryptoGraf - Send Message Keep Secret
Disponible en la URL: http://cryptograf.com/index.php

[3] Java Servlet and JavaServer Pages Technology
Disponible en las URLs: http://java.sun.com/products/servlet/, http://java.sun.com/products/jsp/index.jsp

[4] JDOM - Java-based solution for accessing, manipulating, and outputting XML data from Java code. Disponible en la URL: http://www.jdom.org/

[5] KeyTool IUI - The cryptography GUI tool
Disponible en la URL: http://yellowcat1.free.fr/index_ktl.html

[6] Lleida.net - Chat Systems And Networks
Disponible en la URL: http://lleida.net

[7] Maquina virtual Dalvik
Disponible en la URL: http://www.dalvikvm.com/

[8] Movistar - Servicio Firma Móvil
Disponible en la URL: http://www.movistar.es/public/controller/0,2193,8887_153545600_153545603_0_0,00.html

[9] Open Handset Alliance
Disponible en la URL: http://www.openhandsetalliance.com/

[10] OpenSSL: The Open Source toolkit for SSL/TLS
Disponible en la URL: http://www.openssl.org/

[11] Plataforma Android. Disponible en las URLs: http://code.google.com/intl/es-ES/android/, http://www.android.com/

[12] Public Key Infrastructure
Disponible en la URL: www.sun.com/blueprints/0801/publickey.pdf

[13] Public Key Infrastructure (PKI) and other Concepts in Cryptography
Disponible en la URL: http://www.packtpub.com/article/public-key-\-infrastructurepki-other-concepts-cryptography-cissp

[14] Joan Daemen and Vincent Rijmen, ”The Design of Rijndael: AES - The Advanced Encryption Standard.” Springer-Verlag, 2002. ISBN 3-540-42580-2.

[15] The Legion of the Bouncy Castle. Disponible en la URL: http://www.bouncycastle.org/specifications.html

[16] Using AES with Java Technology - Unlimited strength cryptography. Disponible en la URL: http://java.sun.com/developer/technicalArticles/Security/AES/AES_v1.html

[17] Vodafone - Servicio Firma Electrónica Móvil
Disponible en la URL: http://www.vodafone.es/empresas/servicios/firmaelectronica

Máxima Seguridad para Firmas Digitales con Verificación Distribuida

Javier Herranz[1], Alexandre Ruiz[1], Germán Sáez[1]

***Abstract*—Una de las opciones para proteger el nivel de anonimato o privacidad de un firmante es construir firmas digitales con verificación distribuida: se requiere la colaboración de un subconjunto autorizado de usuarios para verificar la (in)validez de una firma. En RECSI'08, se propuso un esquema de este tipo, pero que no alcanzaba el máximo nivel de seguridad. En este trabajo proponemos el primer esquema de firma digital con verificación distribuida que consigue seguridad máxima, en términos de infalsificabilidad y privacidad. Demostramos formalmente estas dos propiedades por reducción a problemas computacionales estándar, en el modelo del oráculo aleatorio.**

***Index Terms*—Firma digital, compartición de secretos, modelo del oráculo aleatorio, indistinguibilidad**

I. Introducción

En algunas situaciones la propiedad de verificación universal en una firma digital puede ser no deseable, si el firmante desea un cierto nivel de anonimato o de privacidad. Una posible solución a este problema consiste en exigir la colaboración de varios usuarios para que el protocolo de verificación se pueda ejecutar correctamente. Este tipo de esquemas recibe el nombre de esquemas de firma con verificación distribuida, que pueden aplicarse en situaciones reales como subastas o votaciones electrónicas.

En [6], se definen las propiedades de seguridad (infalsificabilidad y privacidad) que debe satisfacer un esquema de firma con verificación distribuida. También se propone un esquema concreto, pero dicho esquema no alcanza el máximo nivel de seguridad respecto a la propiedad de privacidad.

En este trabajo proponemos el primer esquema de firma con verificación distribuida que satisface las máximas propiedades de seguridad. En particular, el esquema es seguro incluso ante atacantes que conocen las claves secretas de todos los participantes (excluido el participante que se está atacando). Conviene remarcar que esta propiedad (conocida como *insider security*, en inglés) no es en absoluto fácil de conseguir: incluso construcciones genéricas obtenidas al combinar un esquema de firma con un esquema de cifrado con descifrado distribuido no satisfacen este nivel máximo de seguridad. La definición detallada de estas propiedades de seguridad se puede encontrar en la Sección III. El diseño del nuevo esquema, que se presenta en la Sección IV, sigue las ideas del esquema de cifrado distribuido de Shoup y Gennaro [9]. En la Sección V, demostraremos formalmente las dos propiedades de seguridad, en el modelo del oráculo aleatorio, por reducción a dos problemas estándar: el problema del logaritmo discreto, y el problema Computacional de Diffie-Hellman.

[1]MAIV, UPC, Barcelona, Spain,
{jherranz,aruiz,german}@ma4.upc.edu

II. Esquemas de firma con verificación distribuida

Un esquema Σ de firma con verificación distribuida consiste en cuatro protocolos probabilísticos y de tiempo de ejecución polinómico:

1) ***Ini*.** La entrada es un parámetro de seguridad λ. Las salidas son unos parámetros públicos $\texttt{params}$ utilizados en todo el esquema.
$$\Sigma.\mathsf{Ini}(1^\lambda) = \texttt{params}$$
2) ***Gen_Cla*.** Este protocolo utiliza dos algoritmos. El primero corresponde al firmante A que obtendrá un par de claves (sk_A, pk_A), donde sk_A es la clave privada para firmar y pk_A es la correspondiente clave pública. El segundo algoritmo corresponde a un conjunto $\mathcal{B}$ de n verificadores, que tiene asociada una estructura de acceso (monótona creciente) $\Gamma_\mathcal{B} \subset 2^\mathcal{B}$, que contiene los subconjuntos autorizados a verificar. Estos usuarios obtendrán cierta información privada $\{sk_j\}_{j\in\mathcal{B}}$ que va a ser usada más tarde en el proceso de verificación distribuida, y cierto valor público $pk_\mathcal{B}$ común para el conjunto $\mathcal{B}$. El proceso de generación de claves para el colectivo $\mathcal{B}$ puede ser ejecutado por una tercera autoridad de confianza o de manera conjunta por ellos mismos, usando técnicas conocidas [3].
$$\Sigma.\mathsf{GC}(\texttt{params}, A, \text{`individual'}) = (sk_A, pk_A)$$
$$\Sigma.\mathsf{GC}(\texttt{params}, \mathcal{B}, \Gamma_\mathcal{B}, \text{`colectivo'}) = (\{sk_j\}_{j\in\mathcal{B}}, pk_\mathcal{B})$$
3) ***Firm*.** Este algoritmo es ejecutado por el firmante A; toma como entrada un mensaje m, su clave privada sk_A y la clave pública asociada a un grupo $\mathcal{B}$ de verificadores, y da como salida una firma $\theta(m)$ del mensaje.
$$\Sigma.\mathsf{Firm}(\texttt{params}, m, pk_\mathcal{B}, sk_A) = \theta(m)$$
4) ***Ver_Dist*.** Dado $B \in \Gamma_\mathcal{B}$ un subconjunto autorizado de verificadores, este protocolo toma como entrada un mensaje m, una firma θ, la clave pública pk_A y los fragmentos sk_j de los usuarios $j \in B$. La salida será 1 si $\theta(m)$ es una firma válida de m y 0 en el caso contrario.
$$\Sigma.\mathsf{Ver}(\texttt{params}, m, \theta, pk_A, B, \{sk_j\}_{j\in B}) = 1 \text{ ó } 0$$

Hay que remarcar que el primer y segundo protocolos se ejecutan sólo una vez. Los otros dos protocolos, i.e. el proceso de firma y de verificación, se ejecutan tantas veces como los participantes quieran firmar o verificar.

III. Modelo de seguridad

Las propiedades de seguridad que un esquema de firma con verificación distribuida deberá satisfacer son las de *infalsificabilidad* y *privacidad*. Nuestro objetivo es considerar las nociones de seguridad más fuertes posibles. Por esta razón al adversario se le permite hacer peticiones de firma y verificación para diferentes usuarios, mensajes y firmas.

Además se le permite corromper al mayor número de participantes posibles (con la excepción de los usuarios que sean objetivo de su ataque en cada caso). En particular, la infalsificabilidad se alcanza incluso cuando el adversario conoce toda la información secreta de todos los participantes con la excepción del firmante que quiere atacar. Por otra parte, la privacidad se consigue incluso en el caso que el adversario conozca las claves secretas de todos los posibles firmantes y de un subconjunto no autorizado de verificadores. Este nivel de seguridad recibe en inglés el nombre de *insider security*.

A. Infalsificabilidad

La *infalsificabilidad existencial contra ataques de mensaje escogido* [5] requiere que cualquier atacante debe tener probabilidad despreciable [1] de falsificar una firma válida de un usuario (del cual no conoce su clave secreta), incluso si el atacante puede obtener previamente otros pares (mensaje,firma) válidos, para mensajes y conjuntos de verificadores que él escoge adaptativamente.

Esta propiedad se formaliza con el siguiente juego, en el que dado un parámetro de seguridad λ un retador externo reta a un atacante $\mathcal{F}_{\mathsf{INF}}$ para que intente falsificar una firma válida nueva:

1) El retador ejecuta $\mathtt{params} \leftarrow \Sigma.\mathsf{Ini}(1^\lambda)$ y da todos los valores obtenidos junto con una estructura de acceso Γ a $\mathcal{F}_{\mathsf{INF}}$.
2) $\mathcal{F}_{\mathsf{INF}}$ escoge un participante $A^\star$ para ser atacado. El retador ejecuta $(sk_{A^\star}, pk_{A^\star}) \leftarrow \Sigma.\mathsf{GC}(\mathtt{params}, A^\star, \text{'individual'})$, se guarda $sk_{A^\star}$ y le da $pk_{A^\star}$ a $\mathcal{F}_{\mathsf{INF}}$.
3) [Generación de nuevas claves] El atacante puede ejecutar $(sk_A, pk_A) \leftarrow \Sigma.\mathsf{GC}(\mathtt{params}, A, \text{'individual'})$ para firmantes $A \neq A^\star$ de su elección, y también puede ejecutar $\Sigma.\mathsf{GC}(\mathtt{params}, \mathcal{B}, \Gamma_{\mathcal{B}}, \text{'colectivo'}) = (\{sk_j\}_{j\in\mathcal{B}}, pk_{\mathcal{B}})$ para conjuntos $\mathcal{B}$ de su elección.
4) [Peticiones hash] Si la seguridad se considera en el modelo del oráculo aleatorio [1], $\mathcal{F}_{\mathsf{INF}}$ puede hacer peticiones al oráculo que modela el comportamiento de ciertas funciones hash.
5) [Peticiones firma] $\mathcal{F}_{\mathsf{INF}}$ puede escoger, de manera adaptativa, tuplas $(m_\ell, pk_{\mathcal{B}_\ell})$ y enviarlas a un oráculo de firma para el firmante $A^\star$. $\mathcal{F}_{\mathsf{INF}}$ obtiene como respuesta las firmas $\theta(m_\ell) \leftarrow \Sigma.\mathsf{Firm}(\mathtt{params}, m_\ell, pk_{\mathcal{B}_\ell}, sk_{A^\star})$.
6) [Falsificación] En un cierto momento, $\mathcal{F}_{\mathsf{INF}}$ publica un par $(m^\star, \theta^\star)$ y una clave $pk_{\mathcal{B}^\star}$ para un conjunto $\mathcal{B}^\star$ y una estructura de acceso $\Gamma_{\mathcal{B}^\star}$. El atacante $\mathcal{F}_{\mathsf{INF}}$ gana el juego si $(m^\star, \theta^\star) \neq (m_\ell, \theta(m_\ell))$, para toda firma obtenida durante el ataque, y además $\Sigma.\mathsf{Ver}(\mathtt{params}, m^\star, \theta^\star, pk_{A^\star}, B, \{sk_j\}_{j\in B}) = 1$, para algún subconjunto $B \in \Gamma_{\mathcal{B}^\star}$.

La ventaja de un adversario $\mathcal{F}_{\mathsf{INF}}$ en romper la infalsificabilidad de un esquema de firma con verificación distribuida se define como

$$\mathsf{Vent}_{\mathcal{F}_{\mathsf{INF}}}(\lambda) = \Pr[\mathcal{F}_{\mathsf{INF}} \text{ gana el juego}].$$

Definición 1: Un esquema de firma con verificación distribuida Σ es infalsificable si para cualquier adversario $\mathcal{F}_{\mathsf{INF}}$ de tiempo polinómico, el valor $\mathsf{Vent}_{\mathcal{F}_{\mathsf{INF}}}(\lambda)$ es despreciable con respecto al parámetro de seguridad λ.

B. Privacidad

Intuitivamente, en una firma digital con verificación distribuida se requiere que un atacante que corrompa a un subconjunto de usuarios no autorizado, no pueda obtener ninguna información sobre la (in)validez de las firmas calculadas por el usuario A. Para formalizar exactamente qué quiere decir 'no obtener ninguna información', se adapta el concepto de seguridad semántica (inicialmente introducido para esquemas de cifrado y conocido también como *indistinguibilidad contra ataques de cifrado escogido* [4] o IND-CCA). De manera informal, dados dos mensajes escogidos por un atacante, y una firma válida para uno de estos mensajes, el adversario no debe ser capaz de distinguir qué mensaje ha sido firmado con probabilidad significativamente mayor que $1/2$ (respuesta aleatoria).

Para formalizar esta idea intuitiva, detallamos aquí un juego de indistinguibilidad donde un atacante $\mathcal{F}_{\mathsf{IND}}$ intenta ganar a un retador externo:

1) El retador ejecuta $\mathtt{params} \leftarrow \Sigma.\mathsf{Ini}(1^\lambda)$ y da todos los valores obtenidos a $\mathcal{F}_{\mathsf{IND}}$.
2) $\mathcal{F}_{\mathsf{IND}}$ escoge un conjunto de verificadores $\mathcal{B}^\star$, una estructura de acceso $\Gamma_{\mathcal{B}^\star} \subset 2^{\mathcal{B}^\star}$ y un subconjunto no autorizado $\widetilde{B} \notin \Gamma_{\mathcal{B}^\star}$, cuyos usuarios puede corromper. El retador ejecuta el protocolo $(\{sk_j\}_{j\in\mathcal{B}^\star}, pk_{\mathcal{B}^\star}) \leftarrow \Sigma.\mathsf{GC}(\mathtt{params}, \mathcal{B}^\star, \Gamma_{\mathcal{B}^\star}, \text{'colectivo'})$, da al atacante $\mathcal{F}_{\mathsf{IND}}$ los valores $pk_{\mathcal{B}^\star}$ y $\{sk_j\}_{j\in\widetilde{B}}$, y mantiene el resto de valores sk_j en secreto.
3) [Generación de nuevas claves] El atacante puede ejecutar $(sk_A, pk_A) \leftarrow \Sigma.\mathsf{GC}(\mathtt{params}, A, \text{'individual'})$ para firmantes A de su elección, y también puede ejecutar $\Sigma.\mathsf{GC}(\mathtt{params}, \mathcal{B}, \Gamma_{\mathcal{B}}, \text{'colectivo'}) = (\{sk_j\}_{j\in\mathcal{B}}, pk_{\mathcal{B}})$ para parejas $(\mathcal{B}, \Gamma_{\mathcal{B}}) \neq (\mathcal{B}^\star, \Gamma_{\mathcal{B}^\star})$ de su elección.
4) [Peticiones hash] Si la seguridad se considera en el modelo del oráculo aleatorio, $\mathcal{F}_{\mathsf{IND}}$ puede hacer peticiones al oráculo que modela el comportamiento de ciertas funciones hash.
5) [Peticiones verificación] $\mathcal{F}_{\mathsf{IND}}$ escoge diferentes tuplas $(m_\ell, \theta_\ell, pk_{A_\ell})$ para firmantes A_ℓ de su elección y hace peticiones, de manera adaptativa, a un oráculo

[1] Formalmente, decimos que una función f es *despreciable* (o *negligible*, en inglés) en k si existe un polinomio $p(\cdot)$ y un valor entero positivo k_0 tal que $f(k) \leq 1/p(k)$ para todo $k \geq k_0$. Usualmente, se escribe $f(k) = \mathrm{negl}(k)$ para las funciones f despreciables en k.

de verificación para estas firmas, con conjunto de verificadores $\mathcal{B}^\star$. $\mathcal{F}_{\mathsf{IND}}$ obtiene como respuesta toda la información emitida durante la ejecución del protocolo $\Sigma.\mathsf{Ver}(\mathtt{params}, m_\ell, \theta_\ell, pk_{A_\ell}, \mathcal{B}^\star, \{sk_j\}_{j\in\mathcal{B}^\star})$.

6) $\mathcal{F}_{\mathsf{IND}}$ escoge dos mensajes m_0, m_1 de la misma longitud y un firmante $A^\star$ con claves $(sk_{A^\star}, pk_{A^\star})$, que $\mathcal{F}_{\mathsf{IND}}$ envía al retador.
7) [Desafío] El retador escoge un bit aleatorio $b \in \{0,1\}$ y ejecuta $\theta^\star \leftarrow \Sigma.\mathsf{Firm}(\mathtt{params}, m_b, pk_{\mathcal{B}^\star}, sk_{A^\star})$. La firma resultante $\theta^\star$ se envía a $\mathcal{F}_{\mathsf{IND}}$.
8) [Más peticiones] Los pasos 4 y 5 son repetidos, con la restricción que las tuplas $(m_i, \theta^\star, pk_{A^\star})$ no pueden ser enviadas al oráculo de verificación, para $i = 0, 1$.
9) Finalmente, $\mathcal{F}_{\mathsf{IND}}$ devuelve un bit $b' \in \{0,1\}$.

Decimos que $\mathcal{F}_{\mathsf{IND}}$ gana el juego si $b' = b$. La *ventaja* de un tal adversario $\mathcal{F}_{\mathsf{IND}}$ en romper la privacidad de un esquema de firma con verificación distribuida se define como

$$\mathsf{Vent}_{\mathcal{F}_{\mathsf{IND}}}(\lambda) = |2\Pr[b' = b] - 1|\,.$$

Definición 2: Un esquema de firma con verificación distribuida Σ satisface la propiedad de privacidad si para cualquier adversario $\mathcal{F}_{\mathsf{IND}}$ de tiempo polinómico, el valor $\mathsf{Vent}_{\mathcal{F}_{\mathsf{IND}}}(\lambda)$ es despreciable con respecto al parámetro de seguridad λ.

IV. El esquema propuesto

En esta sección se describe un esquema específico de firma con verificación distribuida. El diseño de este nuevo esquema sigue las ideas del esquema de cifrado distribuido propuesto por Shoup y Gennaro [9]. Demostraremos que el nuevo esquema satisface las nociones máximas de seguridad de infalsificabilidad y privacidad. Para simplificar y por falta de espacio, consideramos el escenario donde los verificadores actúan correctamente en el proceso distribuido. Una modificación simple de nuestro esquema, incluyendo pruebas no interactivas de conocimiento cero sobre la igualdad de dos logaritmos discretos, permite añadir robustez al esquema para detectar participantes deshonestos en el proceso distribuido de verificación.

Detallamos a continuación los protocolos que componen nuestro esquema de firma con verificación distribuida Σ, para un firmante A y un conjunto $\mathcal{B} = \{1, \ldots, n\}$ de n verificadores.

1) ***Ini.*** $\Sigma.\mathsf{Ini}(1^\lambda)$.
Dado un parámetro de seguridad $\lambda \in \mathbb{N}$, se escogen dos números primos p y q tales que $|q| = \lambda$ y $q|(p-1)$. Se escoge también un grupo cíclico $\mathbb{G} = \langle g\rangle$ de orden primo q. Se escoge otro parámetro κ de seguridad, suficientemente grande (por ejemplo $\kappa = 160$) para evitar ataques de colisión al esquema. Posteriormente se escogen y publican cuatro funciones hash $H_0 : \{0,1\}^* \to \{0,1\}^\ell$, $H_1 : \{0,1\}^* \to \mathbb{G}$, $H_2 : \{0,1\}^* \to \mathbb{Z}_q$ y $H_3 : \{0,1\}^* \to \{0,1\}^\kappa$. En la demostración de seguridad, supondremos que las funciones hash H_0, H_1, H_2 se comportan como un oráculo aleatorio [1]. Del protocolo se obtienen los valores $\mathtt{params} = (p, q, \mathbb{G}, g, \ell, H_0, H_1, H_2, H_3)$.
2) ***Gen_Cla.*** $\Sigma.\mathsf{GC}(\mathtt{params}, A,$ 'individual')
$\Sigma.\mathsf{GC}(\mathtt{params}, \mathcal{B}, \Gamma_\mathcal{B},$ 'colectivo').
Para el firmante A, la clave secreta es un elemento aleatorio $x_A \in \mathbb{Z}_q^*$ que guarda de manera privada, mientras que la clave pública correspondiente es $y_A = g^{x_A} \bmod p$.
Para el colectivo $\mathcal{B}$ de n usuarios se publica el valor $y_\mathcal{B} = g^{x_\mathcal{B}}$ para un valor aleatorio $x_\mathcal{B} \in \mathbb{Z}_q^*$ que es desconocido para los miembros de $\mathcal{B}$. Cada verificador j de $\mathcal{B}$ recibe un fragmento s_j del secreto $x_\mathcal{B}$, correspondiente a un esquema de compartición de secretos de espacio vectorial [2] para la estructura de acceso $\Gamma_\mathcal{B}$. Es decir, para cada conjunto autorizado $B \in \Gamma_\mathcal{B}$ existen coeficientes $\{\lambda_j^B\}_{j\in B}$ tales que $\sum_{j\in B} \lambda_j^B s_j = x_\mathcal{B}$.
3) ***Firm.*** $\Sigma.\mathsf{Firm}(\mathtt{params}, m, y_\mathcal{B}, x_A)$.
Si A quiere firmar un mensaje $m \in \{0,1\}^*$, ejecuta los siguientes pasos:
   a) Escoge un valor aleatorio $r \in \mathbb{Z}_q^*$ y calcula $R = g^r \bmod p$.
   b) Calcula $k = H_0(R, y_\mathcal{B}, (y_\mathcal{B})^r, y_A)$ y $c = H_3(k, m)$.
   c) Elige valores aleatorios $\alpha_1, \alpha_2 \in \mathbb{Z}_q^*$ y calcula $Y_1 = g^{\alpha_1} \bmod p$, $Y_2 = g^{\alpha_2} \bmod p$.
   d) Calcula $\bar{g} = H_1(c, R, Y_1, Y_2, y_A, Y_\mathcal{B}) \in \mathbb{G}$, y posteriormente $\bar{R} = \bar{g}^r \bmod p$, $\bar{Y}_1 = \bar{g}^{\alpha_1} \bmod p$.
   e) Calcula $h = H_2(c, R, \bar{g}, \bar{R}, Y_1, Y_2, \bar{Y}_1, y_A, y_\mathcal{B})$.
   f) Calcula $s_1 = \alpha_1 - h \cdot r \bmod q$.
   g) Calcula $s_2 = \alpha_2 - h \cdot x_A \bmod q$.
   h) Devuelve la firma $\theta(m) = (c, R, \bar{R}, h, s_1, s_2)$.
4) ***Ver_Dist.*** $\Sigma.\mathsf{Ver}(\mathtt{params}, m, \theta, y_A, B, \{s_j\}_{j\in B})$.
Si los participantes de un subconjunto autorizado $B \in \Gamma_0$ quieren cooperar para verificar la firma $\theta(m) = (c, R, \bar{R}, h, s_1, s_2)$ del mensaje m, ejecutan los siguientes pasos.
   a) Cada verificador $j \in B$ calcula $\bar{g} = H_1(c, R, g^{s_1} \cdot R^h, g^{s_2} \cdot (y_A)^h, y_A, y_\mathcal{B})$ y comprueba entonces que la siguiente igualdad se verifica: $h = H_2(c, R, \bar{g}, \bar{R}, g^{s_1} \cdot R^h, g^{s_2} \cdot (y_A)^h, \bar{g}^{s_1} \cdot \bar{R}^h, y_A, y_\mathcal{B})$
   b) Si la igualdad no se verifica, j devuelve $(j, 0)$.
   c) En caso contrario, $j \in B$ devuelve el valor $T_j = R^{s_j} \bmod p$.
   d) Una vez se han recibido valores válidos T_j, diferentes de $(j, 0)$, correspondientes al subconjunto autorizado $B \in \Gamma_\mathcal{B}$, se recupera el valor $R^{x_\mathcal{B}}$: $\prod_{j\in B} T_j^{\lambda_j^B} = R^{x_\mathcal{B}} \bmod p$, donde $\lambda_j^B \in \mathbb{Z}_q$ son los coeficientes definidos por el esquema de compartición de secretos de espacio vectorial.
   e) Se calcula $k = H_0(R, y_\mathcal{B}, R^{x_\mathcal{B}}, y_A)$.
   f) Finalmente, para verificar la validez de la firma se comprueba la igualdad $c = H_3(k, m)$ devolviendo 1 si la igualdad es válida y 0 en caso contrario.

Intuitivamente, dada una firma $\theta(m) = (c, R, \bar{R}, h, s_1, s_2)$,

los dos primeros elementos (c, R) son un código de autenticación del mensaje (MAC, en inglés) para el mensaje m, que puede ser verificado sólo si suficientes miembros de $\mathcal{B}$ cooperan. El resto de elementos $(\bar{R}, h, s_1, s_2)$ corresponden a una prueba de conocimiento cero del logaritmo discreto de y_A y de la igualdad entre los logaritmos discretos $\mathsf{LogDisc}_g(R) = \mathsf{DiscLog}_{\bar{g}}(\bar{R})$. La verificación de dicha prueba de conocimiento se realiza en los pasos a) y b) del protocolo *Ver_Dist*.

V. Análisis de seguridad

En este apartado demostramos que el esquema de firma con verificación distribuida propuesto en la sección anterior satisface las propiedades definidas en la Sección III, alcanzando pues el máximo nivel de seguridad que se puede exigir a este tipo de firmas.

La seguridad se considera en el modelo del oráculo aleatorio [1], donde el atacante puede hacer peticiones al oráculo que modela el comportamiento de ciertas funciones de hash.

Basaremos la seguridad del esquema propuesto en los siguientes problemas computacionales. Dado un parámetro de seguridad $\lambda \in \mathbb{N}$, un número primo q de λ bits y un grupo cíclico $\mathbb{G} = \langle g \rangle$ de orden primo q:

- El problema del *Logaritmo Discreto* (LD): para una entrada $(\mathbb{G}, y)$, tal que $y \in \mathbb{G}$, el objetivo del algoritmo $\mathcal{A}^{LD}$ es encontrar un entero $x \in \mathbb{Z}_q^*$ tal que $y = g^x$. Para un algoritmo en tiempo polinómico $\mathcal{A}^{LD}$ que recibe la tupla $(\mathbb{G}, y)$, definimos $\mathsf{Vent}_{\mathcal{A}^{LD}}(\lambda)$ como la probabilidad de que $\mathcal{A}$ encuentre ese valor $x \in \mathbb{Z}_q^*$ tal que $y = g^x$. La *hipótesis del Logaritmo Discreto* asume que el problema LD es difícil de resolver, es decir que $\mathsf{Vent}_{\mathcal{A}^{LD}}(\lambda)$ es depreciable en λ.
- El problema *Computacional de Diffie-Hellman* (CDH): para una entrada $(\mathbb{G}, g, g^a, g^b)$, tal que $a, b \in \mathbb{Z}_q^*$ son valores aleatorios, el objetivo del algoritmo $\mathcal{A}^{CDH}$ es calcular el valor de $g^{ab} \in \mathbb{G}$. Definimos $\mathsf{Vent}_{\mathcal{A}^{CDH}}(\lambda)$ y la *hipótesis Computacional de Diffie-Hellman* de manera análoga al problema LD.

La infalsificabilidad del nuevo esquema se basará en la dificultad de resolver el problema LD, mientras que la privacidad se basará en la dificultad de resolver el problema CDH.

A. Infalsificabilidad

Antes de proceder con la demostración de infalsificabilidad, recordamos una simplificación del *Forking Lemma*, introducido por Pointcheval y Stern en [7].

Lema 1: [Forking Lemma modificado] Consideremos un esquema de firma digital genérico [2] *con parámetro de seguridad λ. Dado un algoritmo $\mathcal{B}$ que obtiene una firma válida (m, R, h, s) con probabilidad al menos $\varepsilon(\lambda)$, existe otro algoritmo $\mathcal{B}'$ que utiliza $\mathcal{B}$ como subrutina y que produce con probabilidad $\varepsilon'(\lambda) \geq \mathcal{O}\left(\varepsilon(\lambda)^2\right)$ dos firmas válidas (m, R, h, s) y (m, R', h', s') tales que $h \neq h'$.*

[2]Las firmas genéricas tienen la forma (m, R, h, s), donde R es un valor aleatorio escogido dentro de un conjunto muy grande (de tamaño exponencial en el parámetro de seguridad), y $h = H(m, R)$ para una función de hash H.

A continuación demostramos por reducción que nuestro esquema de firma con verificación distribuida es infalsificable, en el modelo del oráculo aleatorio [1], basándonos en la suposición que el problema del logaritmo discreto es computacionalmente irresoluble en grupos de orden primo.

Teorema 1: Sea $\lambda \in \mathbb{N}$ un parámetro de seguridad. Para cualquier atacante $\mathcal{F}_{\mathsf{INF}}$ contra la infalsificabilidad de nuestro esquema de firma con verificación distribuida, existe un algoritmo $\mathcal{A}^{LD}$ para el problema del logaritmo discreto, esencialmente con el mismo tiempo de ejecución que $\mathcal{F}_{\mathsf{INF}}$, tal que

$$\mathsf{Vent}_{\mathcal{A}^{LD}}(\lambda) \geq \mathcal{O}\left(\mathsf{Vent}_{\mathcal{F}_{\mathsf{INF}}}(\lambda)^2\right).$$

Proof: Asumiendo que tenemos un atacante $\mathcal{F}_{\mathsf{INF}}$ que tiene ventaja $\mathsf{Vent}_{\mathcal{F}_{\mathsf{INF}}}(\lambda)$ en romper la infalsificabilidad de nuestro esquema de firma, vamos a construir un algoritmo $\mathcal{A}^{LD}$, que va a ir ejecutando a su vez el atacante $\mathcal{F}_{\mathsf{INF}}$ como subrutina, simulando su entorno y respondiendo a sus peticiones. Aplicando el Lema 1 al atacante $\mathcal{F}_{\mathsf{INF}}$, el algoritmo $\mathcal{A}^{LD}$ será capaz de resolver el problema del logaritmo discreto.

$\mathcal{A}^{LD}$ recibe como entrada un grupo cíclico $\mathbb{G} = \langle g \rangle$ de orden primo q, junto con un valor $y \in \mathbb{G}$. El objetivo de $\mathcal{A}^{LD}$ es encontrar un entero $x \in \mathbb{Z}_q$ tal que $y = g^x$.

Inicialización de $\mathcal{F}_{\mathsf{INF}}$. El protocolo $\Sigma.\mathsf{Ini}(1^\lambda)$ es ejecutado por $\mathcal{A}^{LD}$: éste da a $\mathcal{F}_{\mathsf{INF}}$ los valores $\mathtt{params} = (p, q, \mathbb{G}, g, \ell, H_0, H_1, H_2, H_3)$. Aquí las funciones hash $H_0 : \{0,1\}^* \to \{0,1\}^\ell$, $H_1 : \{0,1\}^* \to \mathbb{G}$ y $H_3 : \{0,1\}^* \to \{0,1\}^\kappa$ son elegidas arbitrariamente por $\mathcal{A}^{LD}$. Sin embargo, H_2 es modelada como un oráculo aleatorio y por ello $\mathcal{A}^{LD}$ mantendrá una tabla TAB_2 que servirá para responder a las peticiones hash de $\mathcal{F}_{\mathsf{INF}}$.

Para simular la ejecución del protocolo $\Sigma.\mathsf{GC}(\mathtt{params}, A^\star, \text{`individual'})$, para el firmante $A^\star$ escogido por $\mathcal{F}_{\mathsf{INF}}$, el algoritmo $\mathcal{A}^{LD}$ define la clave pública de $A^\star$ como $y_{A^\star} = y$ y se la envía a $\mathcal{F}_{\mathsf{INF}}$. Nótese que la correspondiente clave secreta $x_{A^\star}$, que es desconocida por $\mathcal{A}^{LD}$, es precisamente la solución buscada al problema del Logaritmo Discreto.

Generación de nuevas claves. El atacante $\mathcal{F}_{\mathsf{INF}}$ puede generar libremente nuevas claves públicas y secretas para otros firmantes $A \neq A^\star$ y para colectivos $(\mathcal{B}, \Gamma_{\mathcal{B}})$ de verificadores de su elección.

Peticiones hash. Como la prueba es en el modelo del oráculo aleatorio para la función hash H_2, $\mathcal{F}_{\mathsf{INF}}$ puede hacer peticiones de esta función aleatoria. Para ello, $\mathcal{A}^{LD}$ crea y mantiene una tabla TAB_2 que responde de la siguiente manera a estas peticiones: la primera vez que se hace una petición, se escoge un valor aleatorio $h \in \mathbb{Z}_q$, se devuelve h a $\mathcal{F}_{\mathsf{INF}}$, y se guardan los valores de la petición junto con el valor devuelto h en TAB_2. Si la misma petición se hace en el futuro, buscamos en la tabla y $\mathcal{A}^{LD}$ responde con el mismo valor h que se encuentra en la salida existente.

Peticiones firma. Cuando $\mathcal{F}_{\mathsf{INF}}$ solicita firmas válidas para mensajes m_ℓ y claves públicas $y_{\mathcal{B}_\ell}$ de su elección, donde

el firmante es $A^\star$ y $\mathcal{B}_\ell$ es el colectivo de verificadores, $\mathcal{A}^{LD}$ simula y devuelve firmas $\theta(m)$, de la siguiente manera:

1) Elige un valor aleatorio $r \in \mathbb{Z}_q^*$ que le sirve para calcular $R = g^r \bmod p$, $k = H_0(R, y_{\mathcal{B}_\ell}, (y_{\mathcal{B}_\ell})^r, y_{A^\star})$ y $c = H_3(k, m_\ell)$.
2) Elige valores aleatorios $h, s_1, s_2 \in \mathbb{Z}_q$ y calcula $Y_1 = g^{s_1} \cdot R^h \bmod p$, $Y_2 = g^{s_2} \cdot (y_{A^\star})^h \bmod p$.
3) Calcula $\bar{g} = H_1(c, R, Y_1, Y_2, y_{A^\star}, y_{B_\ell})$, y posteriormente $\bar{R} = \bar{g}^r \bmod p$, $\bar{Y}_1 = \bar{g}^{s_1} \cdot \bar{R}^h \bmod p$.
4) Si la entrada $(c, R, \bar{g}, \bar{R}, Y_1, Y_2, \bar{Y}_1, y_{A^\star}, y_{B_\ell})$ se encuentra en TAB_2 (lo que ocurre con probabilidad despreciable), se vuelve al paso 2.
5) $\mathcal{A}^{LD}$ 'falsifica' el oráculo aleatorio para H_2, imponiendo la relación $h = H_2(c, R, \bar{g}, \bar{R}, Y_1, Y_2, \bar{Y}_1, y_{A^\star}, y_{B_\ell})$ en TAB_2. Más tarde, si $\mathcal{F}_{\mathsf{INF}}$ llamara al oráculo aleatorio con esa misma entrada, se le devolvería el valor h.
6) Devuelve la firma $\theta(m_\ell) = (c, R, \bar{R}, h, s_1, s_2)$ a $\mathcal{F}_{\mathsf{INF}}$.

Es fácil comprobar que la firma devuelta es consistente, si no existen colisiones a la hora de 'falsificar' el oráculo aleatorio.

FALSIFICACIÓN. En algún momento $\mathcal{F}_{\mathsf{INF}}$ produce con probabilidad $\mathsf{Vent}_{\mathcal{F}_{\mathsf{INF}}}(\lambda)$ una clave $y_{\mathcal{B}^\star}$ y una firma falsificada $(m^\star, \theta^\star)$ para un conjunto de verificadores $(\mathcal{B}^\star, \Gamma_{\mathcal{B}^star})$, donde $\theta^\star = (c^\star, R^\star, \bar{R}^\star, h^\star, s_1^\star, s_2^\star)$ verifica las siguientes dos propiedades. Primero, la firma $(m^\star, \theta^\star)$ debe ser diferente a las solicitadas anteriormente durante las peticiones de firma y segundo, se debe verificar $\Sigma.\mathsf{Ver}(\mathtt{params}, m^\star, \theta^\star, y_{A^\star}, B, \{s_j\}_{j\in B}) = 1$, para algún subconjunto $B \in \Gamma_{\mathcal{B}^\star}$.

Puesto que la firma falsificada es válida, obtenemos que $h = H_2(c^\star, R^\star, \bar{g}^\star, \bar{R}^\star, Y_1^\star, Y_2^\star, \bar{Y}_1^\star, y_{A^\star}, y_{\mathcal{B}^\star})$, donde $Y_1^\star = g^{s_1^\star} \cdot (R^\star)^{h^\star}$, $Y_2^\star = g^{s_2^\star} \cdot (y_{A^\star})^{h^\star}$, $\bar{Y}_1^\star = (\bar{g}^\star)^{s_1^\star} \cdot \left(\bar{R}^\star\right)^{h^\star}$.

Además, puesto que esta falsificación es diferente de las firmas obtenidas durante las peticiones de firma, podemos estar seguros que la entrada $\mathsf{petición}^\star = (c^\star, R^\star, \bar{g}^\star, \bar{R}^\star, Y_1^\star, Y_2^\star, \bar{Y}_1^\star, y_{A^\star}, y_{B^\star})$ para H_2 no ha sido 'falsificada' y añadida a TAB_2 por $\mathcal{A}^{LD}$.

REPLICANDO EL ATAQUE. Ahora podemos aplicar las técnicas de replicación del Forking Lemma modificado, descrito en el enunciado del Lema 1. De manera informal, $\mathcal{A}^{LD}$ repetirá la ejecución del adversario $\mathcal{F}_{\mathsf{INF}}$, con la misma aleatoriedad pero cambiando los valores salida del oráculo aleatorio H_2 para la entrada $\mathsf{petición}^\star$.

De esta manera, después que $\mathcal{A}^{LD}$ ejecute dos veces a $\mathcal{F}_{\mathsf{INF}}$, obtendremos con probabilidad cuadrática en $\mathsf{Vent}_{\mathcal{F}_{\mathsf{INF}}}(\lambda)$ (la probabilidad de obtener la primera firma falsificada) dos firmas válidas $\theta^\star = (c^\star, R^\star, \bar{R}^\star, h^\star, s_1^\star, s_2^\star)$ y $\theta'^\star = (c'^\star, R'^\star, \bar{R}'^\star, h'^\star, s'^\star_1, s'^\star_2)$ para los mismos valores $(c^\star, R^\star, \bar{g}^\star, \bar{R}^\star, Y_1^\star, Y_2^\star, \bar{Y}_1^\star, y_{A^\star}, y_{B^\star})$ de H_2 tales que $h \neq h'$.

Puesto que las dos firmas son válidas, tenemos que

$$g^{s_2^\star} \cdot (y_{A^\star})^{h^\star} = Y_2^\star = g^{s'^\star_2} \cdot (y_{A^\star})^{h'^\star},$$

lo cual implica que obtenemos la siguiente relación para el valor inicial $y = y_{A^\star} = \left(g^{s_2^\star - s_2'^\star}\right)^{1/(h'^\star - h^\star)}$.

Por tanto, $\mathcal{A}^{LD}$ devuelve el valor $x = \frac{s_2^\star - s_2'^\star}{h'^\star - h^\star} \bmod q$ y resuelve el problema del logaritmo discreto de y en base g, con probabilidad de éxito $\mathsf{Vent}_{\mathcal{A}^{LD}}(\lambda) \geq \mathcal{O}\left(\mathsf{Vent}_{\mathcal{F}_{\mathsf{INF}}}(\lambda)^2\right)$. ■

B. Privacidad

En el siguiente teorema demostraremos que nuestro esquema de firma con verificación distribuida verifica la propiedad de privacidad definida en la Sección III, por reducción al problema computacional de Diffie-Hellman en grupos de orden primo.

Teorema 2: Sea $\lambda \in \mathbb{N}$ un parámetro de seguridad. Para cualquier atacante $\mathcal{F}_{\mathsf{IND}}$ contra la privacidad de nuestro esquema de firma con verficación distribuida, existe un solucionador $\mathcal{A}^{CDH}$ del problema computacional de Diffie-Hellman, esencialmente con el mismo tiempo de ejecución que $\mathcal{F}_{\mathsf{IND}}$, tal que

$$\mathsf{Vent}_{\mathcal{A}^{CDH}}(\lambda) \geq \mathsf{Vent}_{\mathcal{F}_{\mathsf{IND}}}(\lambda)/2.$$

Proof: Asumiendo que tenemos un atacante $\mathcal{F}_{\mathsf{IND}}$ que tiene ventaja $\mathsf{Vent}_{\mathcal{F}_{\mathsf{IND}}}(\lambda)$ en romper la privacidad de nuestro esquema de firma, vamos a construir un algoritmo $\mathcal{A}^{CDH}$ que irá usando a su vez a $\mathcal{F}_{\mathsf{IND}}$ como subrutina, para resolver el problema Computacional de Diffie-Hellman.

El algoritmo $\mathcal{A}^{CDH}$ recibe como entrada un grupo cíclico $\mathbb{G} = \langle g \rangle$ de orden primo q, junto con una tupla (g, g^a, g^b). El objetivo de $\mathcal{A}^{CDH}$ es calcular g^{ab}.

INICIALIZACIÓN DE $\mathcal{F}_{\mathsf{IND}}$. El adverdario $\mathcal{F}_{\mathsf{IND}}$ escoge un conjunto de verificadores $\mathcal{B}^\star = \{1, \ldots, n\}$, una estructura de acceso $\Gamma_{\mathcal{B}^\star} \subset 2^{\mathcal{B}^\star}$ y un subconjunto de participantes corruptos $\widetilde{B} \notin \Gamma_{\mathcal{B}^\star}$.

$\mathcal{A}^{CDH}$ simula una ejecución del protocolo $\Sigma.\mathsf{Ini}(1^\lambda)$: da a $\mathcal{F}_{\mathsf{IND}}$ los valores $\mathtt{params} = (p, q, \mathbb{G}, g, \ell, H_0, H_1, H_2, H_3)$, cuyas funciones hash H_0, H_1 y H_2 las modelará como oráculos aleatorios, mientras que la función $H_3 : \{0,1\}^* \to \{0,1\}^\kappa$ la define explícitamente. Por tanto, $\mathcal{A}^{CDH}$ mantendrá tres tablas TAB_0, TAB_1 y TAB_2 que servirán para responder a las peticiones hash de $\mathcal{F}_{\mathsf{IND}}$.

La ejecución del protocolo $\Sigma.\mathsf{GC}(\mathtt{params}, \mathcal{B}^\star, \Gamma_{\mathcal{B}^\star}, \text{'colectivo'})$ es simulada por $\mathcal{A}^{CDH}$ de la siguiente manera: los fragmentos de los participantes corruptos, $\{s_j\}_{j\in\widetilde{B}}$, son elegidos aleatoria e independientemente en $\mathbb{Z}_q$ y dados posteriormente a $\mathcal{F}_{\mathsf{IND}}$. En este punto, $\mathcal{A}^{CDH}$ define la clave pública $y_{B^\star} = g^b$ que también se envía a $\mathcal{F}_{\mathsf{IND}}$. Remarcamos que esto significa que la clave secreta $x_{B^\star}$ asociada está definida implícitamente como b).

GENERACIÓN DE NUEVAS CLAVES. El atacante $\mathcal{F}_{\mathsf{IND}}$ puede generar libremente nuevas claves públicas y secretas para firmantes A y para colectivos $(\mathcal{B}, \Gamma_{\mathcal{B}}) \neq (\mathcal{B}^\star, \Gamma_{\mathcal{B}^\star})$ de verificadores de su elección.

PETICIONES HASH. Como la prueba es en el modelo del oráculo aleatorio para las funcioness hash H_0, H_1 y H_2, $\mathcal{A}^{CDH}$ crea y mantiene tres tablas TAB_0, TAB_1, y TAB_2 que

simulan estas funciones hash. Para responder a las peticiones hash solicitadas por $\mathcal{F}_{\mathsf{IND}}$, el atacante $\mathcal{A}^{CDH}$ comprueba si ya existe una entrada en la correspondiente tabla para la entrada de esa petición. En ese caso, la salida existente es enviada a $\mathcal{F}_{\mathsf{IND}}$. En caso contrario, una nueva salida es elegida aleatoriamente para ser enviada a $\mathcal{F}_{\mathsf{IND}}$. Posteriormente, la relación entre la entrada y la salida es añadida a la correspondiente tabla.

Para peticiones de H_1, $\mathcal{A}^{CDH}$ elige un valor aleatorio $\beta \in \mathbb{Z}_q^*$ y devuelve el valor $\bar{g} = (g^b)^\beta$ como nueva salida de H_1. El valor β es guardado como valor adicional de la nueva entrada en la tabla TAB_1.

En el caso que $\mathcal{A}^{CDH}$ reciba peticiones de H_0 cuyos primeros valores sean g^a y g^b, el tercer elemento de la entrada será guardado en una tabla adicional $\mathsf{TAB}^\star$. La tabla $\mathsf{TAB}^\star$ será la respuesta final de $\mathcal{A}^{CDH}$ al problema CDH.

PETICIONES VERIFICACIÓN. Cuando $\mathcal{F}_{\mathsf{IND}}$ solicita una petición de verificación $(m_\ell, \theta_\ell, y_{A_\ell})$ para firmantes A_ℓ, con $\theta_\ell = (c_\ell, R_\ell, \bar{R}_\ell, h_\ell, s_{1\ell}, s_{2\ell})$, $\mathcal{A}^{CDH}$ comprueba la validez de la prueba de conocimiento cero $(\bar{R}_\ell, h_\ell, s_{1\ell}, s_{2\ell})$. Para ello, obtiene mediante una petición hash el valor $\bar{g}_\ell = H_1(c_\ell, R_\ell, g^{s_{1\ell}} \cdot {R_\ell}^{h_\ell}, g^{s_{2\ell}} \cdot (y_{A_\ell})^{h_\ell}, y_{A_\ell}, y_{B^\star}) = (g^b)^{\beta_\ell}$ y verifica que se cumpla $h_\ell = H_2(c_\ell, R_\ell, \bar{g}_\ell, \bar{R}_\ell, g^{s_{1\ell}} \cdot {R_\ell}^{h_\ell}, g^{s_{2\ell}} \cdot (y_{A_\ell})^{h_\ell}, \bar{g}_\ell^{s_{1\ell}} \cdot \bar{R}_\ell^{h_\ell}, y_{A_\ell}, y_{B^\star})$. En el caso que no se cumpla, la respuesta de $\mathcal{A}^{CDH}$ a la petición será 0.

En caso contrario, $\mathcal{A}^{CDH}$ debe calcular los valores $\{R_\ell^{s_j}\}_{j \in \mathcal{B}^\star}$ y enviárselos a $\mathcal{F}_{\mathsf{IND}}$. Para los participantes corruptos $j \in \widetilde{B}$, $\mathcal{A}^{CDH}$ puede calcular estos valores fácilmente ya que conoce $\{s_j\}_{j \in \widetilde{B}}$. Además, como es válida la prueba de conocimiento de la igualdad de los logaritmos $\mathsf{LogDisc}_g(R_\ell) = \mathsf{LogDisc}_{\bar{g}_\ell}(\bar{R}_\ell)$, siendo $\bar{g}_\ell = g^{b\beta_\ell}$, tenemos que $R_\ell^{b\beta_\ell} = \bar{R}_\ell$. Por tanto, $\mathcal{A}^{CDH}$ puede calcular $R_\ell^{x_{B^\star}} = R_\ell^b = \bar{R}_\ell^{1/\beta_\ell}$. Conociendo $R_\ell^{x_{B^\star}}$ y $\{R_\ell^{s_j}\}_{j \in \tilde{B}}$, el algoritmo $\mathcal{A}^{CDH}$ puede usar en el exponente las relaciones lineales entre fragmentos y secretos, determinadas por el esquema para compartir secreto que realize $\Gamma_{\mathcal{B}^\star}$, y obtener el resto de valores $\{R_\ell^{s_j}\}_{j \in \mathcal{B}^\star \setminus \widetilde{B}}$.

Finalmente, $\mathcal{A}^{CDH}$ ejecuta una petición hash $H_0(R_\ell, y_{B^\star}, R_\ell^{x_{B^\star}}, y_{A_\ell}) = k_\ell$, actualizando la tabla TAB_0 en caso que esta petición sea nueva, y verifica si $H_3(k_\ell, m_\ell) = c_\ell$. Según el resultado de esa verificación, $\mathcal{A}^{CDH}$ devuelve la salida 1 (firma válida) o 0 (firma no válida).

DESAFÍO. En un momento dado, $\mathcal{F}_{\mathsf{IND}}$ escoge y publica dos mensajes m_0, m_1 de la misma longitud, junto con un firmante $A^\star$ con valores $(x_{A^\star}, y_{A^\star})$. Ahora $\mathcal{A}^{CDH}$ debe enviar una firma $\theta^\star$ a $\mathcal{F}_{\mathsf{IND}}$, que obtiene de la siguiente manera: primero define $R^\star = g^a$ y elige valores aleatorios $c^\star \in \{0,1\}^\ell$, $h^\star, s_1^\star, s_2^\star \in \mathbb{Z}_q$ y $\beta^\star \in \mathbb{Z}_q^*$. Con estos valores $\mathcal{A}^{DH}$ define $\bar{g}^\star = g^{\beta^\star}$, $\bar{R}^\star = (g^a)^{\beta^\star}$, $Y_1^\star = g^{s_1^\star} \cdot (R^\star)^{h^\star}$, $Y_2^\star = g^{s_2^\star} \cdot (y_{A^\star})^{h^\star}$ y $\bar{Y}_1^\star = \bar{g}^{s_1^\star} \cdot (\bar{R}^\star)^{h^\star}$.

Si la entrada $(c^\star, R^\star, Y_1^\star, Y_2^\star, y_{A^\star}, y_{B^\star})$ ya existe en TAB_1, o la entrada $(c^\star, R^\star, \bar{g}^\star, \bar{R}^\star, Y_1^\star, Y_2^\star, \bar{Y}_1^\star, y_{i^\star}, D_{i^\star}^{(i^\star)})$ ya existe en TAB_2, entonces $\mathcal{A}^{DH}$ elige nuevos valores aleatorios $c^\star, h^\star, s_1^\star, s_2^\star$ y $\beta^\star$. Por último, las relaciones $\bar{g}^\star = H_1(c^\star, R^\star, Y_1^\star, Y_2^\star, y_{A^\star}, y_{B^\star})$ y $h^\star = H_2(c^\star, R^\star, \bar{g}^\star, \bar{R}^\star, Y_1^\star, Y_2^\star, \bar{Y}_1^\star, y_{A^\star}, y_{B^\star})$ son añadidas a las tablas TAB_1 y TAB_2 respectivamente.

La firma final $\theta^\star = (c^\star, R^\star, \bar{R}^\star, h^\star, s_1^\star, s_2^\star)$ es enviada al atacante $\mathcal{F}_{\mathsf{IND}}$.

MÁS PETICIONES. El atacante $\mathcal{F}_{\mathsf{IND}}$ puede hacer más peticiones hash y de verificación, que son respondidas de igual manera a como ha sido descrito anteriormente.

El único problema podría aparecer cuando $\mathcal{F}_{\mathsf{IND}}$ pidiese una verificación de una firma válida (m, θ, y_A), con $\theta = (c, R, \bar{R}, h, s_1, s_2)$ tal que el valor $\bar{g} = H_1(c, R, g^{s_1} \cdot R^h, g^{s_2} \cdot (y_A)^h, y_A, y_{B^\star})$ coincidiese con el valor $\bar{g}^\star = g^{\beta^\star}$, ya que este último valor no tiene por qué ser de la forma $(g^b)^\beta$. Sin embargo, esto ocurriría únicamente cuando los valores de entrada de H_1 tanto para esta firma como para la firma del desafío fuesen los mismos. Puesto que las pruebas de conocimiento cero son válidas, obtendríamos que los valores $\bar{R}$ son iguales en ambos casos y, por tanto, que los valores h, s_1, s_2, y_A también deberían ser iguales. La conclusión es que la firma solicitada θ sería la misma que la firma $\theta^\star$ del desafío. Como esto está prohibido por definición, nunca nos encontraremos con este caso.

ANÁLISIS FINAL. $\mathcal{F}_{\mathsf{IND}}$ devuelve un bit b' con el que gana el juego con una probabilidad significativamente mayor que $1/2$. Puesto que H_0 se comporta como una función aleatoria, $\mathcal{F}_{\mathsf{IND}}$ puede ganar sólo si $\mathcal{F}_{\mathsf{IND}}$ ha preguntado anteriormente al oráculo aleatorio el valor $H_0(g^a, g^b, g^{ab}, y_{A^\star})$ correspondiente al desafío $\theta^\star$. Por tanto, con una probabilidad no despreciable $\mathsf{Vent}_{\mathcal{F}_{\mathsf{IND}}}(\lambda)/2$, el valor g^{ab} está en la tabla $\mathsf{TAB}^\star$ construida por $\mathcal{A}^{CDH}$, que contiene los candidatos a solución del problema CDH. Tal y como indican los autores de [9], podríamos usar el auto-corrector Diffie-Hellman descrito en [8] para transformar al atacante $\mathcal{A}^{CDH}$ en otro que en vez de devolver toda una lista de candidatos $\mathsf{TAB}^\star$, devuelve únicamente la solución correcta del problema CDH. ■

Cabe remarcar que en contraposición al esquema propuesto en [6], que sólo asolía la privacidad débil, el esquema propuesto aquí es seguro incluso frente a adversarios que pueden hacer peticiones a un oráculo de verificación.

REFERENCES

[1] M. Bellare y P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. *Proceedings of CCS'93*, ACM Press, pp. 62–73 (1993).

[2] E.F. Brickell. Some ideal secret sharing schemes. *Journal of Combinatorial Mathematics and Combinatorial Computing*, **9**, pp. 105–113 (1989).

[3] R. Gennaro, S. Jarecki, H.Krawczyk and T. Rabin. Secure distributed key generation for Discrete-Log based cryptosystems. *Journal of Cryptology*, vol. **4** (1), Springer-Verlag, pp. 51–83 (2007).

[4] S. Goldwasser y S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, **28**, pp. 270–299 (1984).

[5] S. Goldwasser, S. Micali y R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. of Computing* **17** (2), pp. 281–308 (1988).

[6] J. Herranz, A. Ruiz y G. Sáez. Esquemas de firma digital con verificación distribuida. *Actas de la X Reunión Española de Criptología y Seguridad de la Información, RECSI'08*, pp. 209–216 (2008).

[7] D. Pointcheval y J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology* **13** (3), Springer-Verlag, pp. 361–396 (2000).
[8] V. Shoup. Lower bounds for discrete logarithms and related problems. *Proceedings of Eurocrypt'97*, LNCS **1233**, Springer-Verlag, pp. 256–266 (1997).
[9] V. Shoup y R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. *Journal of Cryptology*, vol. **15** (2), Springer-Verlag, pp. 75–96 (2002).

Un Servicio de Firma Digital de Contratos Basado en Servicios Web

Gerard Draper-Gil*, Josep Lluís Ferrer-Gomila†, Llorenç Huguet-Rotger‡, M. Magdalena Payeras-Capellà§
Departamento de Matemáticas e Informática
Universidad de las Islas Baleares
Palma, ctra. Valldemossa Km. 7,5
Email: *gerard.draper@uib.es, †jlferrer@uib.es, ‡l.huguet@uib.es, §mpayeras@uib.es

***Resumen*—El auge del comercio electrónico esta impulsando la migración de procesos y aplicaciones tradicionales (en papel) al modelo electrónico, pese a que algunos aspectos relacionados con el comercio electrónico no estén bien solucionados, como por ejemplo, el Intercambio Equitativo de Valores en la Firma Electrónica de Contratos. Por otra parte, los Servicios Web (*Web Services*) se están convirtiendo en el standard para la interconexión de aplicaciones y procesos en entornos B2B (*Business to Business*). Pero no existen propuestas basadas en Servicios Web para solucionar el Intercambio Equitativo de Valores en la Firma Electrónica de Contratos. Además, las soluciones suelen presentarse desde un punto de vista técnico, olvidando los aspectos legales.**

En este artículo, presentamos una propuesta para la implementación de un servicio de contratación electrónica basado en la tecnología de Servicios Web, que cumple con los requisitos legales y técnicos necesarios; también analizamos la adecuación de ésta a un entorno de aplicación real. La propuesta demuestra que es posible diseñar soluciones para la firma electrónica de contratos, con tecnología de Servicios Web, que cumplan los requisitos legales y técnicos. Finalmente, se pone de manifiesto la utilidad de estos servicios en aplicaciones de comercio electrónico, y las complicaciones que surgen a la hora de trasladar un diseño teórico (el desarrollo de un protocolo), a un entorno real (aplicaciones existentes).

I. Introducción

La utilización de Internet como canal de ventas ofrece grandes ventajas a clientes y consumidores. El constante crecimiento del uso de Internet en el ámbito del comercio electrónico lo atestigua. Pero la seguridad de las transacciones electrónicas sigue suponiendo un obstáculo a su expansión. Aún así, debemos reconocer que se han realizado grandes esfuerzos para crear un marco jurídico adecuado y obtener soluciones desde el punto de vista técnico. Centrándonos en la contratación electrónica, encontramos múltiples contribuciones en la literatura científica, pero ninguna de ellas ha sido capaz de evolucionar hasta conseguir un nivel de estandarización o adquirir un puesto relevante dentro de las soluciones comerciales, por lo que carecemos de estándares y estándares *de facto*.

Además, empresas e instituciones encuentran dificultades a la hora de interoperar con segundas o terceras partes. Históricamente, han existido soluciones que han intentado superar este obstáculo y facilitar la interoperabilidad entre aplicaciones y procesos, por ejemplo EDI (*Electronic Data Interchange*), pero nunca lograron un éxito generalizado. En este ámbito, los Servicios Web se postulan como la tecnología que finalmente será capaz de conseguirlo. De ahí el interés de combinar una fase esencial del comercio electrónico, la fase de contratación, con la tecnología de los Servicios Web.

Desde un punto de vista de seguridad, la firma digital de contratos es un problema de Intercambio Equitativo. Cada parte involucrada en la transacción tiene un elemento que quiere entregar a cambio de otro elemento, pero nadie quiere dar su parte a no ser que obtenga el elemento intercambiado por la otra parte. Como ejemplo, podemos pensar en comprar a través de Internet, donde intercambiamos productos o servicios por un pago. Otra aplicación típica, sería el traslado de contratos basados en papel a contratos digitales, y sus implicaciones: negociación, generación de documentos, etc.

El objetivo de nuestra investigación es proponer soluciones para desarrollar servicios de firma electrónica de contratos basados en tecnología de Servicios Web y protocolos optimistas, que no requieren intervención directa de la TTP (*Trusted Third Party*). En este artículo presentamos una solución que demuestra que es posible diseñar servicios de firma electrónica que cumplan los requisitos técnicos y legales necesarios, y analizamos las implicaciones y requerimientos derivados del desarrollo de aplicaciones de este tipo.

II. Ejemplo de Aplicación: Agencias de Viajes

En la figura 1 tenemos un ejemplo de aplicación de Servicios Web: Integración de aplicaciones B2B en entornos de servicios turísticos. Típicamente, las agencias de viajes cuentan con aplicaciones internas hechas a medida para gestionar sus reservas, peticiones, inventarios, etc. Al permitir que a través de Servicios Web entidades externas puedan acceder a sus procesos internos, como proveedor, les permite ofrecer servicios en tiempo real, incrementa su eficiencia y aumenta sus posibilidades de negocio. Además, como consumidores de servicios ofrecidos por mayoristas, les permite acceder a múltiples proveedores externos incrementando su oferta de servicios y mejorando su competitividad. En un escenario como éste, se podría utilizar un servicio de contratación electrónica para mantener un registro de transacciones, con validez legal, y para conseguir acceso a nuevos proveedores mediante la firma electrónica de contratos.

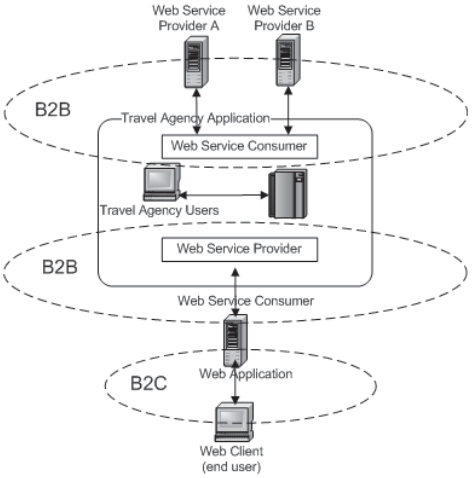


Figura 1. Ejemplo de aplicación

II-A. *Business 2 Business, B2B*

Las agencias de viajes suelen ofrecer servicios de transporte, alojamiento, etc. Para poder adquirir un servicio, el cliente/consumidor tiene que ejecutar una petición de cierre de reserva (*bookingClose*), que típicamente sigue un esquema petición-respuesta. La información incluida en ambos mensajes dependerá de cada tipo de servicio, y de cómo esté implementado. Pero ambos tienen en común el hecho de que la operación de cierre de reserva implica un intercambio equitativo: un usuario intercambia un servicio por un pago. La respuesta del cierre de reserva es el recibo de la transacción, donde el proveedor envía al cliente toda la información sobre el servicio, incluyendo la referencia (localizador) y algunas veces un Bono que se le reclamará al cliente al llegar a destino. En este caso, se podría utilizar el servicio de contratación para mantener un registro de transacciones.

Las agencias suelen ofrecer servicios internos y externos, a través de terceros. Utilizando Servicios Web, una agencia de viajes puede acceder a proveedores externos para mejorar su competitividad y oferta de servicios. Teóricamente, utilizando Servicios Web, una agencia de viajes debería ser capaz de encontrar proveedores de manera automática, dependiendo de la petición. Este descubrimiento y ejecución en tiempo real puede conseguirse si limitamos su aplicación. En lugar de buscar cualquier tipo de proveedor, podemos acotar la búsqueda a proveedores que utilicen un tipo de datos específico, como el que ofrece la OTA (*Open Travel Alliance*). La negociación del contrato puede simplificarse, aceptando un modelo de contrato de antemano. Los problemas de confianza entre usuarios y proveedores puede solventarse con la creación de registros seguros y privados de servicios UDDI (*Universal Description Discovery and Integration*), donde sólo usuarios y proveedores de confianza tengan acceso. Finalmente, podemos utilizar el servicio de contratación electrónica para firmar un contrato/acuerdo con el proveedor y conseguir acceso a sus servicios.

III. Trabajo Previo

Podemos distinguir dos tipos de líneas dentro de la literatura de trabajos previos relacionados con este artículo: los enfocados a proponer soluciones de firma electrónica en entornos de Servicios Web, y los enfocados a la propuesta de soluciones para la contratación electrónica.

III-A. *Firma electrónica en entornos de Servicios Web*

Pese a que existen un gran número de propuestas de protocolo para la firma electrónica de contratos, en lo que se refiere a propuestas de implementación basadas en Servicios Web, el número de publicaciones es sorprendentemente muy bajo. Maruyama *et al*. [1], propone un servicio basado en el protocolo presentado por Asokan *et al*. [2], donde presenta una breve descripción de la implementación del servicio de firma electrónica. Se definen dos estructuras XML (*eXtensible Markup Language*) diferentes para contratos válidos y las estructuras para cada transacción. Los contratos se representan como conjuntos de aserciones en formato SAML (*Security Assertion Markup Language*) y las firmas digitales como estándares XMLdsig (*XML Signature Syntax and Processing*). Aunque la implementación es válida, creemos que al predefinir la estructura de los contratos, limita su posibilidad de aplicación en entornos reales. Robinson *et al*. [3] proponen un servicio de firma electrónica de contratos basado en el protocolo de Zhou y Goldman [4], y Garbinato y Rickebush [6] presentan un protocolo para el intercambio equitativo en entornos de Servicios Web, basada en el uso de módulos hardware a prueba de manipulaciones y enfocada a solucionar el problema de los *Generales Bizantinos*. Ambas propuestas incluyen la utilización de una tercera parte de manera activa (un agente externo y módulos hardware), mientras que la nuestra pretende evitar el uso de terceras partes utilizando protocolos optimistas.

III-B. *Contratación Electrónica*

En el campo de la contratación electrónica, la mayoría de los esfuerzos de investigación están enfocados a resolver el problema desde el punto de vista de procesos de negocio, ignorando o dejando de lado los problemas de seguridad: se intenta automatizar el hecho que una relación contractual entre entidades debe cumplir con una serie de derechos y obligaciones definidos en un contrato, y detectar posibles desviaciones o incumplimientos de éstas. Tienen como objetivo la descripción del conjunto de funcionalidades necesarias para ejecutar el proceso entero, desde la definición/negociación del contrato hasta su ejecución, controlando que no haya desvíos ni incumplimientos. Un buen ejemplo en este campo de investigación, es el artículo de Karlapalem y Krishna [7], donde nos presentan una revisión del trabajo realizado en este ámbito, y proponen una solución para la ejecución del contrato electrónico, sin que los autores hagan referencias específicas a la firma de contratos. En un artículo más reciente, Angelov y Grefen [8] presentan un diseño completo de una arquitectura que cubre el proceso completo de contratación electrónica. Al contrario que en el artículo de Karlapalem y Krishna [7],

en éste sí que encontramos referencias explícitas a la firma electrónica de contratos como parte importante de la fase de Contratación. Define los requisitos de seguridad que el sistema debe cumplir y sus módulos de seguridad. Pero de nuevo falla a la hora de describir cómo se lleva a cabo ese proceso de firma de contratos.

IV. Requisitos

Las bases de nuestra investigación son el marco legal de la firma electrónica, los Servicios Web y los protocolos de firma electrónica de contratos. Cada uno de estos ámbitos tiene sus propios requisitos, por lo que deberemos cumplir con todos ellos.

La firma electrónica de contratos es un caso particular de Intercambio Equitativo, cuyos requerimientos fueron formulados por Asokan *et al*. [2] y más tarde reformulados por Zhou *et al*. [9]: efectividad, equidad, *timeliness*, no-repudio y verificabilidad. Para asegurar el cumplimiento de éstos, utilizaremos la versión simplificada de 2 partes, del protocolo de firma electrónica de contratos presentado por Ferrer-Gomila *et al*. [5]. Se trata de un protocolo optimista en el que se definen 3 sub-protocolos: intercambio, cancelación y finalización. Si las 2 partes implicadas se comportan honestamente, se ejecuta el sub-protocolo de intercambio y no es necesaria la intervención de la TTP; los protocolos de cancelación y finalización se utilizan para resolver posibles conflictos. En la tabla I podemos ver la nomenclatura y elementos que intervienen en el protocolo y en las tablas II, III y IV, la descripción de los 3 sub-protocolos. En el artículo donde se presenta el protocolo (Ferrer-Gomila *et al*. [5]) se puede encontrar un análisis completo sobre la seguridad del protocolo.

Algunas de las razones del éxito de los Servicios Web son, entre otras, su independencia de plataforma y lenguaje de programación, y el hecho de que los Servicios Web se basan en tecnologías de estándares abiertos. Por ello nuestra propuesta también deberá estar basada en estándares abiertos y ser independiente de plataforma y lenguaje. También debemos asegurar que los beneficios derivados del uso de esta tecnología son mayores que los costes, pues de los contrario carecerá de utilidad.

En el artículo de Ferrer-Gomila *et al*. [11] se puede encontrar una discusión sobre los aspectos legales del protocolo de firma electrónica utilizado en la propuesta presentada en este artículo. A destacar, decir que la Directiva de la Union Europea 2000/31/EC [10], con fecha 8 de Junio del 2000 apunta a un sistema de firma electrónica de 3 fases:

1. Una compañía envía una *oferta* utilizando medios electrónicos (un producto o servicio puesto a la venta).
2. Un consumidor (u otra compañía) envía una orden/petición (en relación a la oferta anterior), utilizando medios electrónicos (en términos jurídicos se puede denominar *Aceptación*).
3. La primera compañía debe enviarle al cliente (o segunda compañía) el *acuse de recibo* de la Aceptación que el consumidor le envío.

V. Propuesta de Implementación

V-A. Mensajes y Evidencias

Antes de empezar el diseño de la aplicación, debemos decidir a qué llamamos Mensaje y a qué llamamos Evidencia. Las evidencias son las pruebas que se generan durante la ejecución del protocolo y que permiten a A o B demostrar, en caso necesario, que el contrato ha sido firmado o cancelado. Desde el punto de vista del protocolo es muy sencillo, en las tablas II a IV tenemos definidos para cada paso, el mensaje a enviar. Por ejemplo, en el primer paso del sub-protocolo de intercambio, el mensaje es: M, h_A, y se envía de A a B. Este mensaje incluye una evidencia, h_A, la firma digital de A sobre el contrato M. Pero al intentar trasladar el protocolo a una aplicación real, y en particular a una aplicación basada en Servicios Web, la definición de mensaje y evidencia no es tan sencilla. Los Servicios Web intercambian información en forma de mensajes XML encapsulados en SOAP (*Simple Object Access Protocol*), mientras que las aplicaciones que hacen uso de esos Servicios Web para comunicarse, intercambian mensajes XML. Por ello debemos decidir a qué nivel queremos implementar el protocolo: identificar los mensajes definidos en el protocolo como los mensajes XML intercambiados entre aplicaciones o los mensajes SOAP intercambiados por los Servicios Web. Aunque pueda parecer irrelevante, esta decisión conlleva consecuencias:

- Determinará lo que serán las evidencias y la definición de los tipos XML.
- Afectará al número de estándares, herramientas y tecnologías disponibles para implementar el servicio, como el estándar WSS (*Web Services Security*), que define cómo enviar XML firmados y/o cifrados mediante SOAP.
- Para facilitar la implementación de aplicaciones con Servicios Web, existen varias herramientas que pueden construir Servicios Web de manera automática (Java Axis2, .NET,...), a partir de su descripción en WSDL (*Web Services Description Language*). Si escogemos definir los mensajes como mensajes de aplicación XML, obtendremos una solución a medida, particular para esa aplicación, mientras que si utilizamos los mensajes SOAP como mensajes del protocolo, obtendremos una solución más homogénea. Incluso podemos llegar a encontrar una solución que nos permita describir el servicio de manera estándar y generarlo automáticamente utilizando alguna herramienta.

Para la solución presentada en este artículo, hemos escogido definir los mensajes como los tipos de datos XML intercambiados entre las aplicaciones, para enfocar nuestros esfuerzos en la comprensión de las implicaciones de transformar la *teoría* (descripción de un protocolo) en *práctica* (aplicaciones reales), y su viabilidad.

V-B. Diseño del WSDL

El objetivo final es definir en formato WSDL el servicio de firma digital de contratos. Para ello, lo primero que debemos hacer es definir los tipos de datos, transformar los elementos

M	El contrato a firmar
$h_A = PR_A[H(M)]$	firma digital de A sobre el contrato M
$h_B = PR_B[H(M)]$	firma digital de B sobre el contrato M
$ACK_A = PR_A[H(h_B)]$	firma digital de A sobre h_B. Evidencia de la firma de A.
$ACK_T = PR_T[H(h_B)]$	firma digital de la TTP sobre h_B. Evidencia equivalente a ACK_A
$h_{AT} = PR_A\{[H(H(M), h_A)]\}$	Evidencia que A ha pedido la intervención de la TTP
$h_{BT} = PR_B\{[H(H(M), h_A, h_B)]\}$	Evidencia que B ha pedido la intervención de la TTP
$h_B^{'} = PR_T[H(h_B)]$	firma digital de la TTP sobre h_B como prueba de su intervención
$C_A = PR_T[H(cancelled, h_A)]$	Evidencia de A, como prueba que la firma ha sido cancelada
$C_B = PR_T[H(cancelled, h_B)]$	Evidencia de B, como prueba que la firma ha sido cancelada

Tabla I
NOTACIÓN Y ELEMENTOS UTILIZADOS EN EL PROTOCOLO

del protocolo $M, h_A, h_B, h_B^{'}$, etc. en estructuras de datos XML. Como podemos apreciar en la tabla I, lo que tenemos son un contrato M y firmas digitales. No entraremos en el problema de cómo definir un contrato, puesto que es muy complejo, y entendemos que el contrato debe amoldarse a cada aplicación en particular. Nosotros hemos definido una sencilla estructura con nombre, contenido, fechas, lista de precios, etc. Para las firmas digitales, utilizaremos el estándar *XMLdsig*, junto con *XAdES*, una extensión que cumple con la *Directiva 1999/93/EC del Parlamento Europeo y el Consejo del 13 de Diciembre de 1999*, que establece un marco Europeo para la firma digital. En particular, utilizaremos el formato de firma XML *Detached*, siguiendo las recomendaciones del WS-I (*Web Services Interoperability Organization*) en su documento *Basic Security Profile 1.0*.

Una vez definidos los tipos de datos, debemos definir los mensajes. En la sección IV hemos visto como la normativa Europea apunta a la firma electrónica en 3 pasos, los mismos que tiene el sub-protocolo de intercambio elegido. Así pues, cada paso del sub-protocolo de intercambio será un mensaje distinto. Para los protocolos de Cancelación y Finalización, definiremos un mensaje de petición y uno de respuesta que incluya las 2 posibles respuestas.

Para definir las operaciones debemos analizar el tipo de transacciones ejecutadas en cada sub-protocolo y encontrar su equivalente, si existe, en WSDL. La versión 1.1 de WSDL define 4 tipos posibles (llamados *transaction primitives*):

- *One-way*: El servidor recibe un mensaje.
- *Request-Response*: El servidor recibe una petición y envía su correspondiente respuesta.
- *Solicit-Response*: El servidor envía una petición al cliente, y éste le envía una respuesta.
- *Notification*: El servidor envía un mensaje al cliente.

El sub-protocolo de intercambio se ejecuta en 3 pasos, que no coincide con ninguna de las posibles transacciones definidas para WSDL 1.1, por lo que deberemos describir el sub-protocolo en 2 operaciones distintas. Para que la implementación cumpla con los requerimientos legales vistos en la sección IV, la oferta, primer mensaje, debe enviarlo la compañía o proveedor. Por lo tanto el primer mensaje deberá ser de tipo *Notification* o *Solicit-Response*. Para nuestro diseño, hemos escogido que el paso 1 (oferta) sea una operación

Exchange	Sub-Protocol
A $\rightarrow$ B	M, h_A
A $\leftarrow$ B	h_B
A $\rightarrow$ B	ACK_A

Tabla II
SUB-PROTOCOLO DE INTERCAMBIO

de tipo *Notification*, y los pasos 2 (Aceptación) y 3 (Acuse de recibo de aceptación) sean de tipo *Request*/*Response*. En el caso de los sub-protocolos de cancelación y finalización, deberemos definir una operación de tipo *Request*/*Response* para cada caso.

Hasta este punto hemos realizado una definición abstracta de los servicios, sin especificar dónde se ejecutarán, qué protocolo de transporte se utilizará o cómo se formatearán los mensajes. Para ello, debemos agrupar las operaciones en *PortTypes*, conjuntos de operaciones ejecutados en el mismo punto final (servidor) y especificar, para cada uno de ellos, el protocolo de transporte y formato de mensajes, *Binding*. Al especificar las agrupaciones de operaciones, debemos tener en cuenta que nos interesa mantener separadas las operaciones de cancelación y finalización, de manera que más adelante podamos definir un servicio independiente para cada una de ellas. Aunque las dos operaciones son ejecutadas en el mismo punto final, la TTP, es la entidad A quien ejecutará la operación de cancelación, y la entidad B quien ejecutará la de finalización, tal y como se indica en el protocolo. Por lo tanto, vamos a definir un punto de acceso distinto para cada una de ellas.

Así pues, definiremos 3 agrupaciones de operaciones distintas (*PortTypes*), una para cada servicio: intercambio, cancelación y finalización. Para cada agrupación, especificaremos como protocolo de transporte HTTP y para el formato de mensajes SOAP.

El último paso es definir cada servicio, asignándole un punto de ejecución y agrupaciones de operaciones. Definiremos 3 servicios: intercambio (ejecutado en la entidad A, proveedor), cancelación y finalización (ambos ejecutados en la TTP, pero con distinta URL). El resultado final está disponible en: "http://secom.uib.es/gerard/wsdl.zip".

Cancel Sub-Protocol	
A $\rightarrow$ TTP	$H(M), h_A, h_{AT}$
If finished = true	
TTP	retrieves h_B
TTP $\rightarrow$ A	$h_B, h_B^{'}$
Else cancel signature	
TTP $\rightarrow$ A	C_A
TTP	stores cancelled = true

Tabla III
SUB-PROTOCOLO DE CANCELACIÓN

Finish Sub-Protocol	
B $\rightarrow$ TTP	$H(M), h_A, h_B, h_{BT}$
If cancelled = true	
TTP $\rightarrow$ B	C_B
Else finish signature	
TTP $\rightarrow$ A	ACK_T
TTP	stores finished = true

Tabla IV
SUB-PROTOCOLO DE FINALIZACIÓN

VI. CUMPLIMIENTO DE LOS REQUISITOS

VI-A. Requisitos Técnicos

Los requisitos técnicos que debe cumplir la propuesta de diseño son los enunciados en la sección IV: efectividad, equidad, *timeliness*, no-repudio y verificabilidad.

Hemos diseñado nuestros servicios para que, por cada elemento de información definido en el protocolo (ver tabla I), tenemos un equivalente definido como XML. Es más, si pensamos en cada transacción definida en el protocolo como una operación, podemos comprobar como en el diseño realizado tenemos definidas las mismas operaciones en el mismo orden. Además, las evidencias, firmas digitales en la definición del protocolo, han sido trasladadas a firmas digitales (*XMLdsig*) en el diseño de la implementación. Es más, hemos decidido utilizar la extensión *XAdES* de *XMLdsig*, para cumplir con la Directiva 1999/93/EC del Parlamento Europeo y el Consejo del 13 de Diciembre del 1999, que define un marco común para la firma electrónica. Puesto que el protocolo escogido demuestra cumplir todos los requisitos técnicos (ver Ferrer-Gomila *et al*. [5]), y que la aplicación ha sido diseñada para tener el mismo flujo de ejecución, intercambiar la misma información en el mismo orden, y generar las mismas evidencias, podemos afirmar que nuestra propuesta cumple con las propiedades de efectividad, equidad, *timeliness*, no-repudio y verificabilidad.

VI-B. Requisitos Legales

Al utilizar *XAdES* como formato de firma electrónica, cumplimos con el marco legal de la Union Europea para la firma digital (*Directiva 1999/93/EC del Parlamento Europeo y el Consejo del 13 de Diciembre de 1999*). Además, la propuesta presentada sigue el flujo de ejecución: Oferta-Aceptación-Acuse de recibo, al que apunta la Directiva de la Union Europea 2000/31/EC, con fecha 8 de Junio del 2000 [10], que establece el marco legal de la Unión Europea para la contratación electrónica. Por tanto podemos afirmar que la aplicación cumple con los requisitos legales existentes en la actualidad.

VII. ADECUACIÓN A LAS APLICACIONES

Partiendo de un protocolo optimista (Ferrer-Gomila *et al*. [5]) válido, que ha demostrado cumplir con las características técnicas y requisitos legales (Ferrer-Gomila *et al*. [11]), hemos diseñado una aplicación basada en la tecnología de Servicios Web cumpliendo esos mismos requisitos. En la sección II, hemos presentado un ejemplo de entorno real en el que se aplica la tecnología de los Servicios Web, y se han planteado dos posibles aplicaciones del protocolo para este entorno: un registro de transacciones y una aplicación de firma electrónica de contratos. Ahora nos queda comprobar si la aplicación presentada es apta para utilizarse en la creación de un registro de transacciones y firma electrónica en el entorno propuesto.

VII-A. Registro de transacciones

Como registro de transacciones afectaría a la operación de cierre de reserva *BookingClose*. Se trata de una operación de tipo petición/respuesta, en la que el cliente envía una petición para reservar uno o más servicios. Cuando el proveedor recibe la petición (*BookingCloseRequest*) y confirma su validez, cierra la reserva (la venta se ejecuta) y genera el localizador de reserva o el bono que el consumidor necesitará para poder utilizar el servicio. Esta información se envía al consumidor en la respuesta de cierre de reserva (*BookingCloseResponse*).

- Primero, siguiendo los requisitos legales, el flujo de ejecución del registro de transacciones debe ser: $A \rightarrow B$, $A \leftarrow B$, $A \rightarrow B$, siendo A el proveedor y B el consumidor. Esto implica que la operación de cierre de reserva pasará de tener 2 pasos a tener 4 (el primer mensaje, sera la respuesta del cierre, de $A \rightarrow B$). Duplicar el número de transacciones significa aumentar la probabilidad de que se produzcan errores, disminuir la eficiencia del sistema y aumentar el tiempo de respuesta.
- Segundo, al mapear la respuesta del cierre de reserva con el inicio del registro de transacciones, podemos perder la equidad, y dejar al proveedor en desventaja. Esto es así debido al funcionamiento interno de este tipo de aplicaciones: el proveedor recibe una petición de reserva; si la petición es correcta, procede a cerrar la reserva, bien en su sistema interno (aplicación propia) o bien enviando una petición a su proveedor externo. En cualquiera de los casos, cuando el proveedor responde al cliente, él ya ha ejecutado el cierre de la reserva, lo que quiere decir que si el cliente decidiese echarse atrás, el proveedor debería correr con los gastos de cancelación, en el caso de que los hubiera.
- Tercero, podríamos argumentar que el flujo de ejecución necesario para cumplir los requisitos legales es: $A \rightarrow B$, $A \leftarrow B$, $A \rightarrow B$, siendo A el consumidor y B el proveedor, disminuyendo así el número de pasos a ejecutar, de 4 a 3. Pero volveríamos a perder equidad, en este caso, sería el consumidor el que la perdiese. Si revisamos

el sub-protocolo de intercambio (ver tabla II), vemos que el mensaje que se firma, del que hay evidencias, es M, enviado desde el consumidor al proveedor en el primer paso (asumiendo A = consumidor). El problema viene dado por el hecho de que en la realidad no es así. En la respuesta del cierre de reserva se le envía al cliente el localizador y/o el bono de la reserva, necesarios para poder ejecutarla. Si no conseguimos evidencias que relacionen la transacción con esta información adicional, el proceso es inútil.

Por lo tanto, podemos decir que pese a haber diseñado una aplicación cumpliendo todos los requisitos teóricos y legales necesarios, a la hora de utilizar esta aplicación en un entorno real, que teóricamente encaja en el problema, los resultados no son satisfactorios. Debemos seguir trabajando para acercar las soluciones teóricas, en este caso un protocolo, a las aplicaciones reales y conseguir así las mejoras y beneficios perseguidos con el diseño de esas soluciones.

VII-B. Firma digital de contratos

En el caso de la aplicación de firma digital de contratos propuesta en la sección II, la firma de un contrato implica la ejecución del sub-protocolo de intercambio. Si pensamos en el mundo no electrónico, la identificación de la entidad A como proveedor y la entidad B como consumidor, además de cumplir los requisitos legales, es la lógica. Es el proveedor quien nos propone un contrato ofreciendo sus servicios. Otra posible utilidad de la aplicación presentada en este artículo es la ejecución de la firma digital en un entorno como el presentado en el artículo de Angelov y Grefen [8], donde se define la arquitectura de un sistema de contratación electrónica definiendo un componente específico encargado de la ejecución de la firma.

El único obstáculo que nos podría presentar la aplicación propuesta es el hecho de que el primer paso en la ejecución del sub-protocolo de intercambio está definido como una operación de tipo *Notification*, un mensaje enviado desde el proveedor al consumidor. En la práctica, esto implica que el consumidor debe tener en marcha un servicio para poder atender el mensaje de propuesta de contrato enviado por el proveedor.

VIII. Conclusion

En este artículo hemos presentado una propuesta para la implementación de un servicio de Firma Electrónica de Contratos que cumple con los requisitos técnicos y legales que hemos formulado en la sección IV. Aunque no se trate de una solución estándar, demuestra que es posible diseñar servicios que cumplan ambos requisitos, y nos ayuda a comprender los problemas que surgen al intentar implementar soluciones reales (Firma de Contratos y Registro de Transacciones).

Hemos visto que una aplicación de firma electrónica se puede utilizar para varios propósitos, como registro de transacciones o para ejecutar firmas digitales, como función principal en un proceso de firma electrónica de contratos. Cada usuario tiene diferentes necesidades: como registro de transacciones normalmente requerirá soluciones síncronas, mientras que un servicio de firma digital de contratos puede requerir soluciones asíncronas, debido a la necesidad de intervención humana o de ejecución de procesos que consuman mucho tiempo. El diseño de un servicio basado en mensajes de aplicación XML nos llevará a soluciones hechas a medida para cada usuario, asegurándonos satisfacer todas sus necesidades, pero con un alto grado de incompatibilidad entre aplicaciones de distintos distribuidores. Por otra lado, las implementaciones basadas en mensajes SOAP nos permiten aprovechar estándares existentes como WSS, consiguiendo soluciones más homogéneas.

El siguiente paso en nuestra investigación será la extensión del trabajo a propuestas de protocolos de N-partes y la adecuación de los diseños a las propuestas de aplicación presentadas.

Agradecimientos

Este trabajo ha sido financiado por una beca FPI ligada al proyecto de investigación TSI2007-62986 del Ministerio de Ciencia e Innovación (MICINN) de España, cofinanciada por el Fondo Social Europeo y el proyecto de investigación Consolider, con referencia CSD2007-00004, del MICINN.

Referencias

[1] H. Maruyama, T. Nakamura, T. Hsieh, Optimistic fair contract signing for web services, in: XMLSEC '03: Proceedings of the 2003 ACM workshop on XML security, ACM, New York, NY, USA, 2003, pp. 79–85. doi:http://doi.acm.org/10.1145/968559.968572.

[2] N. Asokan, V. Shoup, M. Waidner, Asynchronous protocols for optimistic fair exchange, in: Security and Privacy, 1998. Proceedings. 1998 IEEE Symposium on, 1998, pp. 86–99. doi:10.1109/SECPRI.1998.674826.

[3] P. Robinson, N. Cook, S. Shrivastava, Implementing fair non-repudiable interactions with web services, in: EDOC '05: Proceedings of the Ninth IEEE International EDOC Enterprise Computing Conference, IEEE Computer Society, Washington, DC, USA, 2005, pp. 195–206. doi:http://dx.doi.org/10.1109/EDOC.2005.16.

[4] J. Zhou, D. Gollmann, Evidence and non-repudiation, J. Netw. Comput. Appl. 20 (3) (1997) 267–281. doi:http://dx.doi.org/10.1006/jnca.1997.0056.

[5] J. L. Ferrer-Gomila, M. Payeras-Capellà, L. H. i. Rotger, Efficient optimistic n-party contract signing protocol, in: ISC '01: Proceedings of the 4th International Conference on Information Security, Springer-Verlag, 2001, pp. 394–407.

[6] B. Garbinato, I. Rickebusch, Orchestrating fair exchanges between mutually distrustful web services, in: SWS '06: Proceedings of the 3rd ACM workshop on Secure web services, ACM, New York, NY, USA, 2006, pp. 33–42. doi:http://doi.acm.org/10.1145/1180367.1180375.

[7] P. R. Krishna, K. Karlapalem, Electronic contracts, IEEE Internet Computing 12 (4) (2008) 60–68. doi:http://doi.ieeecomputersociety.org/10.1109/MIC.2008.77.

[8] S. Angelov, P. Grefen, An e-contracting reference architecture, J. Syst. Softw. 81 (11) (2008) 1816–1844. doi:http://dx.doi.org/10.1016/j.jss.2008.02.023.

[9] J. Zhou, R. H. Deng, F. Bao, Some remarks on a fair exchange protocol, in: PKC '00: Proceedings of the Third International Workshop on Practice and Theory in Public Key Cryptography, Springer-Verlag, 2000, pp. 46–57.

[10] C. European Parliament, Directive 2000/31/ec of the european parliament and of the council of 8 june 2000 on certain legal aspects of information society services, in particular electronic commerce, in the internal market ('directive on electronic commerce'), Official journal l 178 17/07/2000 p. 0001 - 0016.

[11] J. L. Ferrer-Gomila, A. M. Nadal, M. Payeras-Capellà, L. H. i. Rotger, A juridical validation of a contract signing protocol, in: EC-WEB '02: Proceedings of the Third International Conference on E-Commerce and Web Technologies, Springer-Verlag, 2002, pp. 343–352.

On Commitment Schemes Based on Logarithmic Signatures

Pedro Taborda Duarte
Dpto. de Matemática Aplicada
Universidad Rey Juan Carlos
C/ Tulipán s/n. 28933, Móstoles, Madrid, Spain
Email: pedro.duarte@urjc.es

***Abstract*—We consider commitment schemes and covers defined over a non abelian finite group, and propose some lines towards a construction of an unconditionally binding – computationally hiding commitment scheme based on a logarithmic signature.**

I. Introduction

After the advent of public key cryptography, the vast majority of the proposed protocols were based in number theory problems i.e. security would follow from the hardness of solving a certain number theoretic problem. But quantum computing and Shor's algorithms put a date on until when we are unable to solve these number theoretic problems. Hence, research on new computational problems not based un number theory, became important. Recently, some cryptographic protocols were proposed in the area of group theory, using certain factorization sequences (called logarithmic signatures) present in the platform group, where security follows from the assumption that these sequences induce hard factorizations. Some protocols have been proposed using this primitive e.g. MST_1 [11], MST_2 [11] and MST_3 [8], [12] as well as some cryptanalysis [1], [2], [5], [6].

In this work we focus on cryptographic protocols called *commitment schemes* which can be seen as a generalization of encryption schemes. Important examples of commitment schemes include Pedersen's [13] – one of the fastest and most used commitment scheme and based on the well studied discrete logarithmic problem, ElGamal's [4] – one of the simplest and based on the decisional Diffie-Hellman problem, and Groth's [7] – one of the most recent ones, very efficient and based on a not so standard computational problem/assumption called the simultaneous triple pairing assumption.

We propose some lines on how to construct a commitment scheme based on a logarithmic signature in a group, and moreover, committing to group elements. With this, one does not need to convert the committed element to bit strings, as one can, for example, commit to a permutation $\pi \in S_n$ as an ordering of an $n-$set.

II. Commitment Schemes

Commitment schemes are protocols between two parties in which one of the players (the *prover*-**P**) chooses a message m from some (finite) set, and releases some information m_1 to the other player (the *verifier*-**V**). Later, **P** may release more information m_2 to **V** so that he may open this commitment and learn m. Typically, m_1 is some form of m in disguise and m_2 allows to fully reconstruct m from m_1.

An example of a commitment scheme's application are sealed-bid auctions. With it, we achieve secrecy and unambiguity: secrecy to the participants, because the auctioneer cannot learn what is the bid without the bidder's private key (since the bid is locked in a hard to break into safe), making the bid secret until the end of the bidding phase; and unambiguity to the auctioneer because no bidder will be able to change his bid after seeing a previously disclosed opponent's bid (the auctioneer first collects all the safes). The simple functionality of commitment schemes enables the construction of secure protocols that accomplish surprisingly complicated tasks as for example, solving the so called *coin flipping by telephone* problem[1].

A commitment scheme typically has three parts: a *generation key* algorithm, a *commitment* algorithm, and an *opening* algorithm, along with an initial setup of entities where the protocol bases itself. More precisely,

- The key generation algorithm $pk \leftarrow Gen$
- The commitment algorithm $Commit : \mathcal{M} \times \mathcal{R} \rightarrow \mathcal{C} \times \mathcal{D}$, where $\mathcal{M}$ stands for the message space, $\mathcal{R}$ for the space from where **P** introduces *randomness* into the commitment, $\mathcal{C}$ the image space for the commitment values, and $\mathcal{D}$ the space of the *decommitment* values i.e the keys that will allow **V** to "open the box". Most of the times the decommitment values will be of the form $d = (m, r)$ where $m \in \mathcal{M}$ is the committed value and $r \in \mathcal{R}$ the randomness and we will most often not make explicit d when presenting a commitment scheme.
- The *decommiting* algorithm $Open : \mathcal{C} \times \mathcal{D} \rightarrow \mathcal{M} \cup \{\perp\}$. $Open$ will output either an $m \in \mathcal{M}$ if it is presented with a correct commitment/decommitment pair $(c, d) \in \mathcal{C} \times \mathcal{D}$, or $\perp$ if not i.e if **P** doesn't provide **V** with the correct decommiting value d that correctly corresponds to the message m to which the committed value c is attached.

Two basic properties are therefore essential in any commitment scheme: **P** should not be able to change his mind after

[1] see [3] for details

committing to a value, and **V** should not be able to learn about the message committed to. These two properties are referred to as the *binding property* and the *hiding property*. The former ensures fairness to the verifier and the later fairness to the prover.

As in many cryptographic applications, security is based not on an impossibility to break some assumption, but rather on the difficulty to break it. Therefore, regarding the binding and hiding properties, two flavors are considered: *unconditional* and *computational*. Informally, a commitment scheme is computational (resp. unconditional) hiding if any polynomially bounded verifier (resp. any verifier, no matter how powerful) has at most a negligible advantage in finding the committed value (over making a random guess). Computational (resp. unconditional) binding on the other hand means that a polynomially bounded prover (resp. any prover, no matter how powerful), has at most a negligible chance of finding two different messages that commit to the same value.

Definition II.1. *$\epsilon(l)$ is* negligible *in l if for any polynomial f, $\epsilon(l) \leq 1/f(l)$ for all large enough l*

Definition II.2. *Hiding: A commitment scheme is* (t, ϵ)-hiding *if any t-time adversary $A = (A_1, A_2)$ achieves advantage*

$$\left| Pr\left[\begin{array}{l} pk \leftarrow Gen(1^l), \\ s \leftarrow \{0,1\}, \\ (m_0, m_1, \sigma) \leftarrow A_1(pk), \\ (c,d) \leftarrow Commit_{pk}(m_s, r_s) \\ : A_2(\sigma, c) = s \end{array} \right] - \frac{1}{2} \right| \leq \epsilon(l)$$

1) *If t is not infinite i.e if the adversary is polynomially bounded and the advantage is negligible in the length of the input, the scheme is said to be computationally hiding*
2) *If the adversary is infinitely powerful but the advantage is negligible in the length of the input, the scheme is said to be statistically hiding*
3) *If the adversary is infinitely powerful and the advantage is null, the scheme is said to be perfectly hiding*
4) *The scheme is said to be unconditionally hiding if it is perfectly hiding or statistically hiding.*

Definition II.3. *Binding: A commitment scheme is* (t, ϵ)-binding *if any t-time adversary A achieves advantage*

$$Pr\left[\begin{array}{l} pk \leftarrow Gen(1^l), \\ (c, d_0, d_1) \leftarrow A(pk) \\ : \perp \neq Open_{pk}(c, d_0) \wedge \\ \perp \neq Open_{pk}(c, d_1) \wedge \\ Open_{pk}(c, d_0) \neq Open_{pk}(c, d_1) \end{array} \right] \leq \epsilon(l)$$

1) *If t is not infinite i.e if the adversary is polynomially bounded and the advantage is negligible in the length of the input, we speak of computationally binding*
2) *If the adversary is infinitely powerful but the advantage is negligible in the length of the input, the scheme is said to be statistically binding*
3) *If the adversary is infinitely powerful and the advantage is null, we speak of perfectly binding*
4) *The scheme is said to be unconditionally binding if it is perfectly hiding or statistically binding.*

It thus seems that one should aim at building commitment schemes which are both unconditionally hiding and unconditionally binding. Unfortunately, in a two-party commitment scheme, this is impossible: a compromise must be made (see [3]). The main reason for this is that each player sees everything the other party sends. However, in a multi-party scenario, or in a two party case where communication is noisy, it is no longer true that each player sees everything the other party sends, and in these cases it is in fact possible to obtain unconditionally hiding and unconditionally binding commitment schemes.

It can also be proved that, if a commitment scheme is unconditionally binding but its output does not depend on the random parameter, then the scheme cannot be computationally hiding. Hence, in building commitment schemes it is crucial to introduce randomness into the commitment.

III. Covers and Logarithmic Signatures

Most of the well-known public-key cryptosystems which are still unbroken are based on certain intractable problems in large finite abelian groups, such as the multiplicative group of units in the ring $\mathbb{Z}_{pq}$ with p, q primes, the multiplicative group of a finite field, or a cyclic subgroup of the group of rational points of an elliptic curve over a finite field.

One of the first symmetric-key cryptosystems exploiting the structure of non abelian groups was proposed by Magliveras [9] and was named PGM. It explicitly used a special type of factorization in non abelian permutation groups called *logarithmic signatures*, and later the ideas behind it were used in [11] to design the asymmetric schemes MST_1 and MST_2. There are instances using finite groups where it is possible to build a logarithmic signature such that the intractability of solving the discrete logarithmic problem implies the intractability of factoring with respect to the logarithmic signature involved. The idea on using these objects as a tool for building secure cryptosystems, is the assumption of the difficulty to factor elements of the group with respect to the used logarithmic signature. Nevertheless, in subsequent works [2], [6] [2]the assumption of hardness of factoring with respect to a logarithmic signature in these protocols proved to be rather unrealistic by showing that it is unclear on how to generate computationally hard to invert logarithmic signatures for (almost) all group elements. In this work we explore whether they can be useful for designing another cryptographic tool, that of commitment schemes.

A. Definitions and some results

Let G be a finite group and for each $i = 1, \ldots, s$ take $\alpha_i = \{\alpha_{i1}, \ldots, \alpha_{ir_i}\} \subseteq G$ and $S \subseteq G$ a set. Write $\alpha = [\alpha_1, \ldots, \alpha_s]$.

Definition III.1. *Cover and Logarithmic Signature*

1) *Cover: α is said to be a cover for S if any element $g \in S$ can be written as a product*

$$g = \alpha_{1i_1} \dots \alpha_{si_s} \qquad \text{(III-A.1)}$$

The vector $(r_1, \dots, r_s)$ is called the type of the cover and $l = r_1 + \dots + r_s$ its length.

2) *Logarithmic Signature: Let α be a cover for S and let $g \in S$. If the decomposition (III-A.1) is unique, α is said to be a logarithmic signature for S*

Let α as in definition III.1 be a cover of type $(r_1, \dots, r_s)$ for G and let $m = \prod_{i=1}^{s} r_i$. Consider the mappings

$$\lambda : \mathbb{Z}_{r_1} \oplus \dots \oplus \mathbb{Z}_{r_n} \longrightarrow \mathbb{Z}_m$$
$$(k_1, \dots, k_s) \longmapsto \sum_{i=1}^{s} (k_i \prod_{j=1}^{i-1} r_j)$$

and

$$\theta_\alpha : \mathbb{Z}_{r_1} \oplus \dots \oplus \mathbb{Z}_{r_s} \longrightarrow G$$
$$(k_1, \dots, k_s) \longmapsto \alpha_{1k_1} \dots \alpha_{nk_s}$$

Both these mapping are easily seen to be injective and moreover there is an efficient algorithm for computing λ^{-1}. Therefore we are able to efficiently compute [2] $\check{\alpha} = \theta_\alpha \circ \lambda^{-1} : \mathbb{Z}_m \to G$.

Example III.2 (Computation of $\check{\alpha}$)**.** *This is an example taken from [8] and exemplifies the computation of $\check{\alpha}$ in the alternating group $\mathcal{A}_5$. In table I each row of the left column represents a block of α. Let also $\lambda_{\alpha i}$ $i = 1, \dots, 3$ be represented by each row from the right column in table I.*

Then it is easy to check that $\lambda_\alpha = [\lambda_{\alpha 1}, \lambda_{\alpha 2}, \lambda_{\alpha 3}]$ is a logarithmic signature for the additive group $\mathbb{Z}_{60}$. Because $|\mathcal{A}_5| = 60$ we wish to compute $\check{\alpha} : \mathbb{Z}_{60} \to \mathcal{A}_5$ on an element $x \in \mathbb{Z}_{60}$. Such an x can be uniquely written as a sum of elements from λ_α by using a greedy selection algorithm for each component, sequentially from the bottom block upwards. This essentially determines $\lambda^{-1}(x) = (j_1, j_2, j_3)$ with $j_i \in \lambda_{\alpha i} \subseteq \mathbb{Z}_{r_i}$ $r_i = |\lambda_{\alpha i}| \in \{3, 4, 5\}$, and by retrieving the associated element $x_i \in \alpha_i$ of j_i we then set $\check{\alpha}(x) = x_1 \cdot x_2 \cdot x_3$.

If $x = 47$ then x is written uniquely as $47 = 2 + 9 + 36$ and we are looking for $(j_1, j_2, j_3) \in \mathbb{Z}_3 \times \mathbb{Z}_4 \times \mathbb{Z}_5$ such that $47 = j_1 \cdot 1 + j_2 \cdot 3 + j_3 \cdot 12$. It is immediate then to verify that it must be $\lambda(2, 3, 3) = 47$. Therefore, by inspection on the table, we get that

$$\begin{aligned} \check{\alpha}(47) &= \alpha_{1\,2}\alpha_{2\,3}\alpha_{3\,3} \\ &= (1)(2)(354) \cdot (1)(243)(5) \cdot (13)(2)(45) \\ &= (123) \end{aligned}$$

Two logarithmic signatures α and β for G are said to be equivalent if $\check{\alpha} = \check{\beta}$. We now define what is meant by "hard to factor" and "easy to factor" covers (or logarithmic signatures. See also [10])[3].

[2] If α is a logarithmic signature $m = |G|$

α	λ_α
$\mathcal{A}_5$	$\mathbb{Z}_{60}$
(1)(2)(345)	0
(1)(2)(354)	1
(1)(2)(3)(4)(5)	2
(1)(2 3)(4 5)	0
(1)(2 5 3)(4)	3
(1)(2 4 3)(5)	6
(1)(2)(3)(4)(5)	9
(1 2 4)(3)(5)	0
(1)(2 3 5)(4)	12
(1 3)(2)(4 5)	24
(1 5 3 4 2)	36
(1 4 3 2 5)	48

TABLE I
LOGARITHMIC SIGNATURE α AND CANONICAL LOGARITHMIC SIGNATURE λ_α

Definition III.3. *Let G be a finite group of degree n and α a cover for G* [4].

1) *α is said to be tame if there exists a ppt algorithm $\mathcal{A}$ such that $\mathcal{A}(\alpha, g) \in \check{\alpha}^{-1}(g)$ for any $g \in G$*

   i.e if with overwhelming probability, $\mathcal{A}$ outputs one of the several factorizations III-A.1 of element g

2) *α is said to be wild if it is not tame*

   i.e if any ppt algorithm $\mathcal{A}$, has negligible probability in outputting a factorization of a random $g \in G$

Given a cover α and an element $g \in G$, to find one $x \in \check{\alpha}^{-1}(g)$ it is necessary to obtain any of the possible s-tuples $(i_1, \dots, i_s)$ such that $g = \alpha_{1i_1} \dots \alpha_{ni_n}$ and then compute $x = \lambda(i_1, \dots, i_s)$. This is possible if and only if α is tame. Nevertheless, there are instances where obtaining this factorization is indeed believed to be hard:

Example III.4. *In the class $\mathcal{G}$ of finite groups, there are instances (G, α) with α a cover, where the factorization in (III-A.1) is equivalent to solving the discrete logarithmic problem in G.*

Proof: Take the finite field $\mathbb{F}_q$ and consider its multiplicative group $G = \mathbb{F}_q^*$ on which the discrete logarithm problem is believed to be hard. Suppose l is the least positive integer such that $2^{l-1} \leq |G| \leq 2^l$, let $g \in G$ be a generator, and consider the set $\alpha = [\alpha_1, \dots, \alpha_l]$ with $\alpha_i = \{1, g^{2^{i-1}}\}$. Then α is a cover [5] *for G and moreover, factorization with respect to α amounts to solving the discrete logarithm problem in G.* ■

Example III.5. *In the (infinite) group of n-braids B_n with generators $\{\sigma_1, \dots, \sigma_{n-1}\}$ it can be proved that the elements*

$$A_{ij} = (\sigma_{j-1} \dots \sigma_{i+1})\sigma_i^2 (\sigma_{j-1} \dots \sigma_{i+1})^{-1}$$

[3] In [11] the authors proposed a public-key cryptosystem MST_1, whose security is precisely based on the computational difficulty of inverting $\check{\alpha}$ when α is a logarithmic signature

[4] We take $S = G$ although everything can be translated to the case $S \subset G$

[5] α is called a 1-*quasi-logarithmic signature*. See e.g. [11]

for $1 \leq i < j \leq n$ generate the subgroup P_n of pure braids, and that for each $i \in \{1, \ldots, n-1\}$, the set $G_i = \{A_{i,i+1}, \ldots, A_{i,n}\}$ is a free set of generators of a subgroup of P_n which is isomorphic to the free group F_{n-i}. Artin showed that any pure braid $\beta \in P_n$ can be written uniquely as a product

$$\beta = c_1 c_2 \cdots c_{n-1}$$

where each c_i is a unique reduced word in the generators of G_i and their inverses. More precisely, Artin showed that

$$P_n = F_{n-1} \rtimes (F_{n-2} \cdots (F_2 \rtimes F_1))$$

The normal form of this iterated semidirect product is called Artin's combing *of pure braids.*

We can therefore generate a logarithmic signature α of type $(r_1, \ldots, r_{n-1})$ for $S \subsetneq P_n$, by setting

$$\alpha_{i,j} \in < A_{i,i+1}, \ldots, A_{i,n} >$$

for $i = 1, \ldots, n-1$ and $j = 1, \ldots, r_i$.

Remark III.6. *All known algorithms for combing a random pure braid β written in the generators $A_{i,j}$ and their inverses, are not polynomial in general in the length of β and hence we may assume that α is wild in S. Nevertheless, since $c_{n-1} \in \langle A_{n-1,n}^{\pm 1} \rangle$, it is always feasible to find the last block c_{n-1} of β.*

IV. A Commitment Scheme Based on a Logarithmic Signature

We propose a commitment scheme in the line of group theory based cryptography. The scheme's security is based on the hardness of solving a certain problem in a group theoretic setting and commitments are made to group elements not integers/bits. More precisely, for its construction we use a public, hard to invert logarithmic signature over a finite group G, and a public general function $H : (\mathbb{Z}_{|G|})^s \to G$. The use of a logarithmic signature α and its unique factorization property implies almost immediately that the scheme is *unconditionally binding*. Computational hiding follows if H is such that a certain decisional problem - involving α and H - on the base group is hard (definition IV.1). It is a decisional problem related to computational indistinguishability of group elements *à la decisional Diffie-Hellman* problem. If this problem is hard we call G a DLS_H- group.

Let G be a group, and $\alpha = [\alpha_1, \ldots, \alpha_s]$ with $\alpha_i = \{\alpha_{i1}, \ldots, \alpha_{ir_i}\}$, $1 \leq i \leq s$ a *logarithmic signature* for G. If $g = \check{\alpha}(x) = \alpha_{1k_1} \cdots \alpha_{nk_s}$ we write $Bl_i(g) \equiv \alpha_{ik_i}$ $i = 1, \ldots, s$

Note that $\prod_{i=1}^{s} r_i = |G|$ because α is a logarithmic signature. As the security parameter we take the Log of the order of G i.e. $l =_{def} Log|G|$

Definition IV.1. *(H-Decisional) Let $H : (\mathbb{Z}_{|G|})^s \to G$ be some function. G is said to be a $DLS_H - group$ if for any efficient algorithm B, the modulus of the difference between*

$$Pr[(x^1, \ldots, x^s) \leftarrow_R (\mathbb{Z}_{|G|})^s : B((\check{\alpha}(x^1), \ldots, \check{\alpha}(x^s)), H(x^1, \ldots, x^s)) = 1]$$

and

$$Pr[(x^1, \ldots, x^s) \leftarrow_R (\mathbb{Z}_{|G|})^s, \hbar \leftarrow_R G : B((\check{\alpha}(x^1), \ldots, \check{\alpha}(x^s)), \hbar) = 1]$$

is negligible in the security parameter l. This quantity is denoted $Adv_H(B)$.

A. The Scheme

Let $H : (\mathbb{Z}_{|G|})^s \to G$ be a function, and consider the following commitment scheme where the *prover* **P** commits to an element $g \in G$ and sends the commitment to a *verifier* **V**. We can describe it the following way:

Gen(1^l)

1) Generates a logarithmic signature α
2) Generates a function $H : (\mathbb{Z}_{|G|})^s \to G$
3) Publishes $pk = (\alpha, H)$

Commit

To commit to $g \in G$,

1) **P** chooses *u.a.r* an s-tuple $(x^1, \ldots, x^s) \in (\mathbb{Z}_{|G|})^s$
2) **P** computes $c_1 = g \cdot H(x^1, \ldots, x^s)$ and $c_2 = (\check{\alpha}(x^1), \ldots, \check{\alpha}(x^s))$
3) **P** sends **V** the pair (c_1, c_2)

Open

1) **P** sends the committed element g and the randomness $(x^1, \ldots, x^s)$ to **V**
2) **V** checks **P**'s computations and outputs "accept" if they are correct or $\perp$ if not

The following result immediately follows from the uniqueness of factoring through α:

Proposition IV.2. *If α is a logarithmic signature for G then the commitment scheme described above is unconditionally binding.*

Proof: If the prover can find $(g_0, (x^1, \ldots, x^s))$ and $(g_1, (y^1, \ldots, y^s))$ such that

$$Comm(g_0, (x^1, \ldots, x^s)) = Comm(g_1, (y^1, \ldots, y^s))$$

then $\check{\alpha}(x^i) = \check{\alpha}(y^i)$ for all $i = 1, \ldots, s$, and

$$g_0 \cdot H(x^1, \ldots, x^s) = g_1 \cdot H(y^1, \ldots, y^s)$$

Since α is a logarithmic signature we get $x^i = y^i$ for all $i = 1, \ldots, s$, and therefore $g_0 = g_1$. ∎

Proposition IV.3. *If G is a $DLS_H - group$ then the commitment scheme is computationally hiding.*

Proof: The following is the original hiding game between an adversary $\mathcal{A}$ and the challenger (see figure 1):

Game 0

1) *$\mathcal{A}$ chooses and sends the challenger a pair $(g^0, g^1) \in G^2$*

2) *The challenger randomly chooses a bit b and keeps it (i.e does not disclose it to $\mathcal{A}$)*
3) *The challenger randomly chooses $(x^1, \ldots, x^s)$ in $(\mathbb{Z}_{|G|})^s$ and sets $c_2 = (\check{\alpha}(x^1), \ldots, \check{\alpha}(x^s))$*
4) *The challenger sets $c_1 = g^b \cdot H(x^1, \ldots, x^s)$*
5) *The challenger sends $\mathcal{A}$ the pair (c_1, c_2)*
6) *$\mathcal{A}$ outputs a bit $\bar{b}$*

Letting S_0 be the event that $\bar{b} = b$ in GAME 0*, the adversary's advantage is $Adv(\mathcal{A}) \equiv |Pr[S_0] - 1/2|$. $\mathcal{A}$ wins the game if $Adv(\mathcal{A})$ is non negligible in l.*

We now make one small change to GAME 0 *by computing c_1 as $g^b \cdot \hbar$ for a randomly chosen $\hbar \in G$ (see figure 2):*

(g^0, g^1)	$\leftarrow$	$\mathcal{A}$
b	$\leftarrow_R$	$\{0,1\}$
$(x^1, \ldots, x^s)$	$\leftarrow_R$	$(\mathbb{Z}_{\|G\|})^s$
c_1	$\leftarrow$	$g^b \cdot H(x^1, \ldots, x^s)$
c_2	$\leftarrow$	$(\check{\alpha}(x^1), \ldots, \check{\alpha}(x^s))$
$\bar{b}$	$\leftarrow$	$\mathcal{A}(c_1, c_2)$

Fig. 1. GAME 0

GAME 1

1) *$\mathcal{A}$ chooses and sends the challenger a pair $(g^0, g^1) \in G^2$*
2) *The challenger randomly chooses a bit b and keeps it (i.e does not disclose it to $\mathcal{A}$)*
3) *The challenger randomly chooses $(x^1, \ldots, x^s)$ in $(\mathbb{Z}_{|G|})^s$ and sets $c_2 = (\check{\alpha}(x^1), \ldots, \check{\alpha}(x^s))$*
4) *The challenger randomly chooses $\hbar$ in G and sets $c_1 = g^b \cdot \hbar$*
5) *The challenger sends $\mathcal{A}$ the pair (c_1, c_2)*
6) *$\mathcal{A}$ outputs a bit $\bar{b}$*

(g^0, g^1)	$\leftarrow$	$\mathcal{A}$
b	$\leftarrow_R$	$\{0,1\}$
$\hbar$	$\leftarrow_R$	G
$(x^1, \ldots, x^s)$	$\leftarrow_R$	$(\mathbb{Z}_{\|G\|})^s$
c_1	$\leftarrow$	$g^b \cdot \hbar$
c_2	$\leftarrow$	$(\check{\alpha}(x^1), \ldots, \check{\alpha}(x^s))$
$\bar{b}$	$\leftarrow$	$\mathcal{A}(c_1, c_2)$

Fig. 2. GAME 1

Let S_1 be the event that $\bar{b} = b$ in GAME 1.

Claim 1. *$Pr[S_1] = 1/2$. This follows from the fact that on this game, the view of the adversary is independent from b.*

Claim 2. *$Adv(\mathcal{A}) = Adv_H(B)$ is* negligible*. This follows from claim 1. and from observing that in Game 0 the pair (c_1, c_2) is essentially of the form $((\check{\alpha}(x^1), \ldots, \check{\alpha}(x^s)), H(x^1, \ldots, x^s))$ while in Game 1 it is of the form $((\check{\alpha}(x^1), \ldots, \check{\alpha}(x^s)), \hbar)$. To be more precise, consider the following algorithm B on input $((w^1, \ldots, w^s), \hbar) \in G^s \times G$ (see figure 3):*

1) *$\mathcal{A}$ chooses and sends B a pair $(g^0, g^1) \in G^2$*
2) *B chooses a random bit b and keeps it (i.e does not disclose it to $\mathcal{A}$)*
3) *B computes $c_1 = g^b \cdot \hbar$, sets $c_2 = (w^1, \ldots, w^s)$, and sends the pair (c_1, c_2) to $\mathcal{A}$*
4) *$\mathcal{A}$ decides and sends B a bit $\bar{b}$*
5) *If $\bar{b} = b$ then B outputs 1, else outputs 0*

$B((w^1, \ldots, w^s), \hbar)$:

(g^0, g^1)	$\leftarrow$	$\mathcal{A}$
b	$\leftarrow_R$	$\{0,1\}$
c_1	$\leftarrow$	$g^b \cdot \hbar$
c_2	$\leftarrow$	$(w^1, \ldots, w^s)$
$\bar{b}$	$\leftarrow$	$\mathcal{A}(c_1, c_2)$
IF $\bar{b} = b$	THEN	Output 1
ELSE		Output 0

Fig. 3.

If $\hbar = H(x^1, \ldots, x^s)$ with $w^i = \check{\alpha}(x^i)$ $(i = 1, \ldots, s)$ then computation proceeds as in GAME 0 *and therefore*

$$\begin{aligned} &Pr[(x^1, \ldots, x^s) \leftarrow_R (\mathbb{Z}_{|G|})^s : \\ &\quad B((\check{\alpha}(x^1), \ldots, \check{\alpha}(x^s)), H(x^1, \ldots, x^s)) = 1] \\ &\quad = Pr[S_0] \end{aligned}$$

Else it proceeds as in GAME 1 *and therefore*

$$\begin{aligned} &Pr[(x^1, \ldots, x^s) \leftarrow_R (\mathbb{Z}_{|G|})^s, \hbar \leftarrow_R G : \\ &\quad B((\check{\alpha}(x^1), \ldots, \check{\alpha}(x^s)), \hbar) = 1] \\ &\quad = Pr[S_1] \\ &\quad = 1/2 \end{aligned}$$

Hence $Adv(\mathcal{A}) = |Pr[S_0] - Pr[S_1]| = Adv_H(B)$. Therefore, under the DLS_H assumption, the commitment scheme is computationally hiding . ■

V. CONCLUSION

We presented a proposal of a commitment scheme using logarithmic signatures. A logarithmic signature for which the problem of inverting it is hard, provides on its own an idea of computational hiding and perfect binding. The problem is that, using directly the logarithmic signature α (more precisely: the special function $\check{\alpha}$ from the ring $\mathbb{Z}_{|G|}$ to G associated with α. It is this function which is assumed to be hard to invert) it appears we would end up having to commit to strings of bits. One of the objectives here is to build a commitment scheme where the space of messages is a space of elements from a group and not bitwise. Next step in this line of research would perhaps revolve around finding a sufficient condition on the logarithmic signature so as to imply the hardness of the DLS_H

problem (since wildness of α is a necessary condition), or methods for efficiently generating hard to factor logarithmic signatures on G. We feel that logarithmic signatures are a primitive with a large potential and can, in the line of research around group theory based cryptography, pose several interesting problems.

REFERENCES

[1] S. R. BLACKBURN, C. CID, AND C. MULLAN, *Cryptanalysis of the mst_3 public key cryptosystem*, 2009. http://eprint.iacr.org/2009/248.

[2] J. BOHLI, M. GONZÁLEZ VASCO, C. MARTÍNEZ, AND R. STEINWANDT, *Weak Keys in MST_1*, Designs, Codes and Cryptography, 37 (2005), pp. 509–524.

[3] I. DAMGÅRD, *Commitment schemes and zero-knowledge protocols*, in Lectures on Data Security, Modern Cryptology in Theory and Practice, Summer School, Aarhus, Denmark, July 1998, London, UK, 1999, Springer-Verlag, pp. 63–86.

[4] T. ELGAMAL, *A public key cryptosystem and a signature scheme based on discrete logarithms* , IEEE Trans. Info. Theory, 31 (1985), pp. 469–472.

[5] M. GONZÁLEZ VASCO, A. L. P. D. POZO, AND P. T. DUARTE, *A note on the security of mst3*, 2009. preprint: http://eprint.iacr.org/2009/096. Communicated at the "Cryptology, Designs and Finite Groups 2009" meeting, May 17-22, 2009, Deerfield Beach, Florida.

[6] M. GONZÁLEZ VASCO AND R. STEINWANDT, *Obstacles in Two Public-Key Cryptosystems Based on Group Factorizations*, in Cryptology, Tatra Mountains Math. Publications, 2002, pp. no. 3: 23–37.

[7] J. GROTH, *Homomorphic trapdoor commitments to group elements*. Cryptology ePrint Archive, Report 2009/007, 2009. http://eprint.iacr.org/2009/007.

[8] W. LEMPKEN, T. V. TRUNG, S. S. MAGLIVERAS, AND W. WEI, *A public key cryptosystem based on non-abelian finite groups*, J. Cryptol., 22 (2008), pp. 62–74.

[9] S. MAGLIVERAS, *A cryptosystem from logarithmic signatures of finite groups*, in Proceedings of the 29'th Midwest Symposium on Circuits and Systems, Elsevier Publishing Company, 1986, pp. 972–975.

[10] S. MAGLIVERAS AND N. MEMON, *Algebraic properties of cryptosystem PGM*, Journal of Cryptology, 5 (1992), pp. 167–183.

[11] S. MAGLIVERAS, D. STINSON, AND T. V. TRUNG, *New approaches to designing public key cryptosystems using one-way functions and trapdoors in finite groups*, Journal of Cryptology, 15 (2002), pp. 285–297.

[12] S. S. MAGLIVERAS, P. SVABA, T. V. TRUNG, AND P. ZAJAC, *On the security of a realization of cryptosystem $MST3$*, in Cryptology, Tatra Mountains Math. Publications, 2008, pp. no. 3: 65–78.

[13] T. P. PEDERSEN, *Non-interactive and information-theoretic secure verifiable secret sharing*, in CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology, London, UK, 1992, Springer-Verlag, pp. 129–140.

ACKNOWLEDGEMENTS

With support from *Fundação para a Ciência e Tecnologia, Portugal ref: SFRH/BD/37869/2007*. I would like to especially thank Professors María Isabel González Vasco and Ángel Perez del Pozo for their insightful comments.

Implementación de la generación y firma RSA distribuida en procesos de voto electrónico

Alex Escala
Departament d'Enginyeria Telemàtica
Universitat Politècnica de Catalunya
Email: alex.escala@ma4.upc.edu

Sandra Guasch
Scytl Secure Electronic Voting
Email: sandra.guasch@scytl.com

Carlos Luna
Departament d'Enginyeria Telemàtica
Universitat Politècnica de Catalunya
Email: carlos.luna@ma4.upc.edu

***Abstract*—Uno de los pasos más importantes en los procesos de voto electrónico es la firma digital de la lista de votos descifrados, dado que a partir de estos votos se podrán realizar recuentos paralelos para asegurar la validez del resultado de la elección. Para ello, un posible escenario consiste en que los miembros de la Mesa Electoral generen los parámetros de la firma y la ejecuten de forma distribuida. En este artículo proponemos una implementación robusta y eficiente de la generación distribuida de los parámetros RSA y de la firma RSA distribuida.**

I. Introducción

Hoy en día, el uso de las nuevas tecnologías en los procesos electorales es cada vez más común, dadas las numerosas ventajas que proporcionan. La gestión electrónica de unas elecciones permite obtener los resultados finales más rápidamente y realizar un recuento más preciso que si se hiciera a mano. Además permite incorporar dispositivos o métodos especiales que faciliten el ejercicio de su derecho a voto a personas mayores o a discapacitados. En el caso de que se realicen de forma remota, por ejemplo por Internet, los votantes que están en el extranjero en el momento de las elecciones pueden votar de una forma más cómoda y segura que mediante el voto postal.

Sin embargo, la introducción de procesos lógicos que ejecutan los diferentes pasos del proceso electoral proporciona nuevos desafíos a la hora de asegurar la validez del resultado de las elecciones. Existen varios requisitos de seguridad que una elección electrónica debe cumplir para ser como mínimo tan fiable como una elección convencional, como la autenticidad y privacidad de los votantes, la precisión de los resultados de la elección, la verificabilidad del proceso electoral, etc.

Para cumplir algunos de estos requisitos de seguridad es frecuente utilizar herramientas criptográficas estándar como el cifrado y la firma digital de los votos.

En un esquema básico de voto electrónico utilizando estas herramientas, los votantes cifran sus votos en el momento de su emisión y los firman digitalmente usando sus credenciales (por ejemplo el DNIe). Al finalizar la fase de votación, estas firmas se verifican y se separan de los votos cifrados, que pasan por un algoritmo de *anonimización* (para desconectarlos de la identidad de sus emisores) antes de ser descifrados para realizar el recuento y obtener los resultados.

El cifrado de los votos en el momento de su emisión permite preservar la privacidad del votante, ya que un atacante externo que acceda al voto emitido por un votante no podrá descubrir el texto en claro y, por tanto, las opciones de voto que ha escogido. La firma digital permite verificar la integridad del voto (cifrado) e identificar a su autor mediante su certificado digital, comprobando que está en el censo electoral.

Uno de los procesos cuya verificación es más importante en una elección es el correcto recuento de los votos. En una elección llevada a cabo por medios electrónicos esta verificación se realiza de forma similar que en una elección convencional: realizando un recuento paralelo. Es importante asegurar que el conjunto de votos sobre los que se realiza el recuento, es decir, los votos descifrados al final del proceso electoral, es auténtico y no ha sido modificado por ningún intruso malicioso. Para asegurar la integridad de la lista de votos descifrados, ésta se firma digitalmente.

Firma digital. Una firma digital [Sch07] es una herramienta criptográfica que se utiliza para demostrar el origen y la integridad de un mensaje. Así, la validación de una firma digital permite al receptor comprobar que el mensaje ha sido creado por un emisor determinado (autenticidad) y que el mensaje no ha sido alterado durante su tránsito (integridad). Esto puede acometerse con diversos niveles de seguridad. El más exigente de ellos determina que un adversario no puede generar una firma válida para cualquier mensaje aún teniendo a su disposición la firma de cualquier otro mensaje que haya escogido él mismo. Esta noción de seguridad se conoce como no falsificación existencial bajo ataque con mensaje escogido adaptativamente - en inglés, Existential Unforgeability against adaptive Chosen Message Attacks, (EUF-CMA) [GMR88].

En términos generales un esquema de firma digital debe disponer de dos protocolos, uno para generar la firma y otro para verificar su validez. Cuando se usan algoritmos de clave pública para firmar digitalmente un mensaje, normalmente se calcula una función de Hash sobre éste, que es la que se firma.

Firma digital con RSA. Uno de los algoritmos criptográficos de clave pública más utilizados en las firmas digitales, y en el que nos centraremos en este artículo, es RSA [RSA78], dado su amplio uso, su estandarización y la posibilidad de utilizarlo conjuntamente con certificados digitales.

Dados unos parámetros RSA N, e y d, firmar un mensaje M

consiste, simplemente, en realizar la exponenciación modular:

$$Sign(m, d) = m^d \bmod N$$

donde $m = H(M)$ es el Hash mensaje a firmar, d es la clave privada y $N = pq$ es el producto de dos primos de gran tamaño (del orden de 1024 bits cada uno).

Para verificar dicha firma basta con calcular el Hash del mensaje recibido y compararlo con el resultado de otra exponenciación modular sobre la firma, esta vez usando la clave pública e del emisor:

$$Verif(Sign(m, d), e) = (m^d)^e \bmod N \stackrel{?}{=} m \bmod N$$

La Mesa Electoral es el organismo que cuenta con mayor confianza en un proceso electoral. Este organismo posee las claves más importantes de la elección: la clave para descifrar los votos y la clave para firmar la lista de votos descifrados. Dado que algún miembro de la mesa electoral podría actuar de forma fraudulenta, una práctica habitual es utilizar un esquema de compartición de secretos para dividir y repartir las claves entre todos los miembros, de forma que únicamente los subconjuntos autorizados pueden reconstruirlas para descifrar o firmar de forma distribuida un documento.

Esquemas de compartición de secretos. Los esquemas de compartición de secretos tienen como objetivo la división de un secreto en fragmentos de modo que sólo los conjuntos autorizados de usuarios pueden reconstruir el secreto. Cuando se aplica un esquema umbral a estos métodos, el resultado es que se necesitan un mínimo de fragmentos (el umbral) para reconstruir el secreto. Si uno de los poseedores de un fragmento no quiere o no puede participar en la reconstrucción, el secreto se puede reconstruir a partir del resto de fragmentos (si llegan al umbral).

En 1979, Adi Shamir [Sha79] publicó un método para compartir secretos con una estructura de acceso que sigue un esquema de umbral. Su protocolo se basa en generar un polinomio cuyos coeficientes se eligen de manera aleatoria, independiente y uniforme en $\mathbb{Z}_q$ (con q primo):

$$p(x) = d + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$$

donde d es el valor secreto a compartir. El esquema puede modificarse para compartir valores secretos en otro anillo (por ejemplo, $\mathbb{Z}_N$).

Una vez creado el polinomio, cada usuario recibirá un par (i, d_i) donde i es el identificador único de dicho usuario (típicamente: $1, 2, \ldots, n$) y $d_i = p(i)$ es su fragmento del secreto. Por otra parte, el grado de $p(x)$ determina el umbral del esquema de acceso dado que se requieren t evaluaciones de un polinomio de grado $t - 1$ para determinar éste unívocamente. La recuperación del secreto puede hacerse usando los coeficientes de Lagrange:

$$d = p(0) = \sum_{i \in S} d_i \lambda_i^S$$

donde S es un conjunto de t usuarios y λ_i^S, los coeficientes de Lagrange, cumplen $\lambda_i^S = \prod_{\substack{j \in S \\ j \neq i}} \frac{j}{j-i}$

A. Firma RSA distribuida

Dada una clave secreta d compartida con un esquema de umbral en $\mathbb{Z}_N$, cada usuario puede utilizar su fragmento d_i de la clave para realizar una firma parcial del documento, $s_i = m^{d_i}$. La firma m^d puede reconstruirse a posteriori con tan sólo t firmas parciales s_i y sin necesidad de reconstruir explícitamente la clave d.

Los sistemas de compartición de secretos basan su seguridad en el hecho de que no se puede reconstruir la clave sin disponer de un número mínimo de fragmentos. Sin embargo, esta clave ha existido en algún momento antes de distribuir los fragmentos. Para prevenir la posibilidad de que alguien pueda acceder a ella antes de ser destruida, se pueden utilizar algoritmos que generan de forma distribuida los fragmentos sin necesidad de calcular la clave previamente. Estos algoritmos, sin embargo, requieren un gran intercambio de información para garantizar que los cálculos se realizan correctamente. Para ello disponemos de los esquemas de compromiso, que son una primitiva criptográfica de gran importancia en el ámbito de la computación distribuida.

Compromisos de Pedersen. En [Ped91] se presenta un esquema de compromiso orientado a la verificación de secretos compartidos. Dicho esquema permite comprometer un mensaje antes de enviarlo para que el receptor pueda estar seguro de que dicho mensaje no se modificará a posteriori. Así mismo, el emisor del compromiso puede demostrar que todo los fragmentos enviados han sido calculados de acuerdo con el esquema de compartición de secretos acordado y son coherentes entre sí.

Para comprometer un valor $x \in \mathbb{Z}_p$ se genera un elemento aleatorizador $r \in_R \mathbb{Z}_p$ y se calcula:

$$com(x, r) = g^x h^r \bmod p$$

donde $g \in \mathbb{Z}_p$ es un generador del subgrupo de orden q de $\mathbb{Z}_p^*$ (con p y q primos de gran tamaño tales que q divide $p - 1$) y $h = g^\nu$ para algún ν desconocido por los todos usuarios.

Para *abrir* un compromiso de Pedersen es preciso hacer públicos los valores x y r. Sin embargo, disponemos de otras primitivas, las pruebas de conocimiento cero, que nos permiten demostrar que se conocen dichos valores sin necesidad de revelarlos. En [DF02] Damgård y Fujisaki propusieron una versión de los compromisos de Pedersen que es apta para comprometer valores de $\mathbb{Z}$ y que es la que usamos en nuestra implementación.

B. Generación distribuida de parámetros RSA

Los sistemas de generación distribuida de parámetros RSA basan su funcionamiento en la búsqueda de un número producto de dos primos a partir de fragmentos aportados por todos los participantes en el protocolo. Una vez encontrado este número, se crean a partir de él los fragmentos de la clave privada (uno por cada participante) y se calcula la clave pública entre todos.

En este artículo nos centraremos en la implementación práctica de la firma distribuida RSA de la lista de votos descifrados (por parte de los miembros de la Mesa Electoral) utilizando sistemas de compartición de secretos, y en la generación distribuida de los parámetros de la clave privada para realizar esta firma.

El objetivo de esta implementación es obtener una solución que se pueda aplicar y utilizar en entornos reales y actuales, teniendo como punto de partida la literatura existente y el estado del arte, pero sin olvidar las condiciones del entorno en el que se utilizará. Dado que actualmente los componentes de la infraestructura de clave pública trabajan mayoritariamente con el criptosistema RSA (Autoridades Certificadoras, DNIe, etc), nos hemos decantado por el uso de este algoritmo en lugar de otros más ventajosos o eficientes, como DSA o algoritmos basados en la criptografía de curvas elípticas, cuya implantación no está aún debidamente extendida.

Existen ciertos requisitos de seguridad y eficiencia que deben cumplirse en estas implementaciones. Por ejemplo, dado que se firma información muy sensible para la elección, el protocolo deberá ser robusto y seguro: un número de atacantes menor que el umbral de reconstrucción no debe ser capaz de impedir la generación de una firma distribuida correcta, ni averiguar información importante sobre los parámetros RSA generados de forma distribuida. También deberán poder ejecutarse en un tiempo razonable: no se debería exigir la disponibilidad de los miembros de la Mesa un día entero (por ejemplo) para generar los parámetros. La capacidad de firmar de forma distribuida un documento proporciona la posibilidad de que los miembros de la Mesa Electoral actúen de forma remota (si son de regiones lejanas o en caso de enfermedad). Así pues, sería conveniente que el protocolo de firma distribuida fuera lo menos interactivo posible para poder facilitar esta solución, ya que los participantes no tendrían que ir esperándose unos a otros, y podrían ejecutar el protocolo a diferentes horas. Finalmente, debemos considerar que una Mesa Electoral está formada por alrededor de 10 miembros.

El artículo está estructurado de la siguiente forma: en la sección II se presenta una propuesta de implementación de un algoritmo de firma distribuida RSA y en la sección III se explica un método para la generación distribuida y robusta de un módulo RSA.

II. Firma digital distribuida

Como se ha introducido anteriormente, la finalidad del sistema a implementar es firmar un documento de forma distribuida mediante una estructura de umbral. Para hacerlo, cada firmante dispondrá de un fragmento de la clave privada, que habrá sido repartida mediante un esquema de compartición de secretos. La manera de obtener dichos fragmentos se comentará en la sección III. Durante el proceso de firma, cada participante usa su fragmento para obtener una firma parcial que tendrá que ser verificada por los otros participantes.

Una firma digital distribuida tiene ciertos requisitos que deberemos tener en cuenta a la hora de implementar nuestra solución: imposibilidad de falsificación (requisito heredado de la firma digital estándar) y robustez. El primer requisito implica que un grupo de participantes solamente puede firmar un mensaje si es un conjunto autorizado. En el caso de una firma de umbral, un conjunto está autorizado si contiene, al menos, un número determinado de participantes, por ejemplo, más de la mitad. La robustez del sistema es la propiedad que garantiza que, aunque algunos participantes sean maliciosos, se pueda obtener la firma del mensaje. Como requisito adicional, es interesante que se pueda detectar a los participantes que no hayan realizado correctamente el protocolo.

Otra característica a tener en cuenta es la interactividad del sistema de firma digital distribuida. En la medida que sea posible, interesa que el protocolo sea no interactivo. De esta manera, cada participante puede hacer sus cálculos de manera independiente y no debe esperar datos de los otros participantes.

A. Trabajo previo

Shoup [Sho00] es el primero en proponer un esquema eficiente de firma de umbral RSA que es a la vez robusto y no interactivo. Como dice en su artículo, hasta ese momento solamente había sistemas que o eran interactivos, o no eran robustos o generaban firmas de tamaño excesivo.

Para su esquema, Shoup utiliza un módulo RSA N producto de dos primos seguros p y q (decimos que p es un primo seguro si $(p-1)/2 = p'$ también es primo). Para conseguir un protocolo no interactivo usa pruebas de conocimiento cero no interactivas, que resultan muy sencillas gracias a que p y q son primos seguros. El problema de esta propuesta reside en que la generación de un módulo RSA de estas características es mucho más complicada que la generación de un módulo RSA cualquiera (sin restricciones): si para encontrar un primo de k bits necesitamos examinar $O(k)$ números, para encontrar un primo seguro del mismo tamaño necesitamos examinar $O(k^2)$ candidatos.

Posteriormente, en 2005, Damgård y Dupont [DD05] proponen un esquema de firma umbral RSA robusto, sin hipótesis añadidas y con un módulo RSA genérico. Sin embargo, no imponer restricciones al módulo RSA acarrea diversos inconvenientes. El primero de ellos es que las pruebas de conocimiento cero fallan con probabilidad no negligible y que, por lo tanto, es necesario reconstruir la firma diversas ocasiones (con diversas combinaciones de firmas parciales) para asegurar que el protocolo sea robusto y eficiente. El segundo es que las pruebas de conocimiento nulo, y por tanto el protocolo, pasan a ser interactivos.

Para que un esquema de firma umbral RSA sea robusto, el número de participantes maliciosos no puede ser demasiado grande. En concreto, si t es el número de participantes maliciosos y n el número de participantes en total, se tiene que cumplir que $t < n/2$. Este requisito es intuitivo: si queremos que los participantes maliciosos no puedan firmar un mensaje cualquiera (seguridad) y que si se los expulsa igualmente se pueda realizar la firma (robustez), entonces el número de participantes maliciosos tiene que ser menor que el número de participantes buenos.

B. Nuestra implementación

Para realizar nuestra implementación, partimos de la base que tenemos un módulo RSA al que no podemos imponer ninguna restricción. El objetivo es obtener una implementación robusta, segura y, a poder ser, no interactiva.

Nuestro esquema se basa en el protocolo de Damgård y Dupont [DD05] realizando algunas modificaciones para explotar al máximo la idea de hacer distintas combinaciones en la fase de reconstrucción de la firma. En todos los artículos anteriormente citados, se da por supuesto que el número de participantes puede ser arbitrario y, en consecuencia, los autores trataban de hacer que la complejidad creciera controladamente con el número de participantes. En el escenario real en el que se centra nuestro trabajo tendremos un número reducido de participantes (alrededor de 10) y, por tanto, este factor no resulta tan determinante.

En concreto, nuestra propuesta se basa en eliminar las pruebas de conocimiento cero (que fallan con probabilidad no negligible y son interactivas si se usa un módulo RSA cualquiera), consiguiendo un protocolo no interactivo y más efectivo que el esquema de [DD05] en el escenario en que nos encontramos (alrededor de 10 participantes). Si bien es cierto que hay métodos para hacer pruebas de conocimiento cero no interactivas, el coste de éstas las hace impracticables si se usa un módulo RSA sin restricciones, como sucede en nuestro esquema. La robustez, que antes se derivaba de las pruebas de conocimiento cero, la conseguimos mediante repetidas reconstrucciones de la firma. A continuación justificamos por qué el esquema sigue siendo seguro y robusto.

La seguridad es inmediata ya que no depende de las pruebas de conocimiento cero sino del esquema de compartición de secretos y el esquema de firma RSA estándar.

Para justificar la robustez solamente se tiene que ver que el tiempo que tardaremos en obtener la firma correctamente es finito y razonable. Dada la restricción del número de participantes maliciosos, sabemos que al menos una combinación de las firmas parciales será correcta (aquella que provenga de participantes honestos). Por otra parte, en el peor de los casos contaremos con el máximo número de participantes maliciosos y la combinación correcta será la última de las que examinaremos. En total hay $\binom{n}{t+1}$ maneras de combinar las firmas parciales para reconstruir la firma. En la *Tabla I* se expone cuánto vale este número para distintos valores de n (participantes) y tomando t (participantes maliciosos) máxima tal que $t < n - t$. Si t no es máxima, entonces el número de combinaciones posibles es menor que el presentado en la *Tabla I*. En conclusión, el número de combinaciones es razonable para los valores de n que se usarán en la práctica (aunque crece exponencialmente al aumentar n) y por lo tanto podemos obviar las pruebas de conocimiento cero y limitarnos a repetir la fase de reconstrucción hasta obtener una firma válida. Para reducir el número esperado de combinaciones necesarias para encontrar la correcta, se elige aleatoriamente el orden en el que se realizarán las reconstrucciones. De ese modo los participantes maliciosos no podrán forzar el caso peor.

n	$\binom{n}{t}$
3	3
5	10
8	56
10	210
12	792
14	3003

Tabla I: *Maneras de escoger t elementos entre n*

Por otra parte también se ha implementado una subrutina que se encarga de la detección de participantes malignos. En el protocolo de [Sho00] esta detección se realizaba en las pruebas de conocimiento cero. La solución que nosotros proponemos es sencilla y efectiva siempre que tengamos valores de n pequeños. Una vez obtenida la firma, basta con coger el conjunto de participantes honestos y sustituir un participante por otro que no esté en el conjunto. Si el participante nuevo fuera honesto, la firma resultante debería ser correcta, de no ser así el participante es malicioso. En total deberemos hacer $n-t-1$ reconstrucciones extra, que se pueden reducir usando la información que se obtiene al intentar construir la firma.

C. Resultados obtenidos

El estudio teórico de nuestra implementación arroja los siguientes resultados. Como se puede ver en el artículo en que nos basamos [DD05], para realizar una firma parcial cada participante tiene que hacer una sola exponenciación modular. Por otra parte, cada reconstrucción de la firma consta de dos pasos: primero se elevan las $t + 1$ firmas parciales a los coeficientes de Lagrange correspondientes y luego se multiplican entre ellas. Posteriormente se hacen dos exponenciaciones más para conseguir la firma y finalmente se comprueba que la firma es correcta, haciendo una última exponenciación.

Dado que la firma parcial solamente se calcula una vez, se requieren $t + 4$ exponenciaciones que deberán repetirse hasta $\binom{n}{t}$ veces. Aunque puede parecer un número grande (para $n = 10$ son cerca de 1700 exponenciaciones), todos los exponentes excepto el último son números pequeños, de unos 45 bits, y en consecuencia las exponenciaciones son mucho más rápidas que con un exponente arbitrario de 2048 bits. Si comparamos nuestro esquema con el propuesto por Damgård y Dupont, reducimos el número de operaciones en un factor de 10 (y conseguimos que el protocolo sea no-interactivo) en el escenario usual de 10 participantes y claves de 2048 bits.

El resultado también se puede comparar con una firma RSA

estándar. En el caso de hacer una firma RSA distribuida el coste computacional es aproximadamente equivalente a una firma RSA estándar por cada firmante y, como máximo, $\binom{n}{t+1}$ exponenciaciones equivalentes a una firma RSA estándar cada una realizadas por un usuario cualquiera para reconstruir la firma correcta.

Como mejora adicional, los distintos participantes pueden hacer las reconstrucciones en paralelo a costa de perder la propiedad de no-interactividad del protocolo. En este caso el tiempo esperado se reduciría de manera proporcional al número de participantes honestos.

III. Generación distribuida y robusta de un módulo RSA

El protocolo de firma distribuida requiere la existencia de un módulo RSA ($N = pq$) y la distribución de su clave privada (d) entre los participantes siguiendo el esquema de acceso pertinente (en este caso, un esquema de umbral).

En general, este punto se resuelve haciendo uso de una Tercera Parte de Confianza que genera el módulo RSA y la clave secreta y distribuye esta última adecuadamente.

Existen, sin embargo, protocolos para la generación distribuida de módulos RSA que permiten prescindir de la Tercera Parte de Confianza mejorando así la seguridad del sistema. Es por lo tanto interesante obtener un protocolo combinado de Generación y Firma que sea distribuido y robusto.

A. Trabajo previo

Todos los protocolos conocidos hasta la fecha para la generación distribuida de módulos RSA son fuertemente interactivos y, además, requieren un gran número de iteraciones para obtener un módulo válido. En consecuencia, la cantidad de cálculos adicionales que se necesitan para garantizar la robustez del protocolo los hace inviables en escenarios reales.

En 1997 Boneh y Franklin [BF97] presentaron un protocolo para generar módulos RSA de manera distribuida que era seguro contra adversarios pasivos, es decir, aquellos que siguen las especificaciones del protocolo pretendiendo obtener información de los participantes honestos. Sin embargo, no protege de adversarios activos, aquellos que no siguen las especificaciones del mismo. Lamentablemente este nivel de seguridad no concuerda con los estándares de robustez que se han aplicado al protocolo de firma (adversarios activos) y por lo tanto es necesario buscar otra alternativa.

Un año después Frankel, MacKenzie y Yung [FMY98] ofrecieron una versión robusta de dicho protocolo que es, de hecho, la base de nuestra implementación. Su protocolo se compone de tres sub-protocolos, a saber, generación y multiplicación distribuida de p y q, test de biprimalidad y generación de la clave privada d. En ausencia de conductas maliciosas, las dos primeras partes se repiten una y otra vez hasta obtener un módulo RSA válido ($N = pq$ tal que p y q son primos) mientras que la última parte tan sólo se ejecuta una vez. Si, por el contrario, se detectan anomalías en el proceso se procede a identificar al causante (o los causantes) de dichas irregularidades para expulsarlos del proceso y se reinicia el protocolo.

Posteriormente han aparecido diversas alternativas en este campo, como la propuesta de Algesheimer, Camenisch y Shoup [ACS02] y la de Damgård y Mikkelsen [DM10], ambas más ineficientes que la presentada anteriormente. En el primer caso, la ineficiencia proviene de la necesidad de conseguir un módulo RSA formado por primos seguros. En el segundo caso, el test de primalidad de Miller-Rabin que se utiliza, aún finalizando en una única ronda, también es más ineficiente que [FMY98].

B. Nuestra implementación

Como ya se ha mencionado, nuestra implementación se basa en el protocolo de Frankel et al. [FMY98] pero introduce algún cambio de cara a mejorar su eficiencia.

En dicho protocolo cada participante elige dos números aleatorios p_i y q_i y ejecuta un protocolo de multiplicación distribuida para obtener el valor de $N = pq = \sum p_i \cdot \sum q_i$. Acto seguido se aplica un test distribuido de biprimalidad aprovechando la información de la que dispone cada usuario (p_i, q_i y N). Este proceso de generación y chequeo se repite hasta obtener un módulo RSA válido (N es producto de dos primos). La idea general consiste en usar una versión no robusta del protocolo por defecto durante la fase de prueba y error para encontrar el módulo adecuado y, cuando el módulo generado sea producto de dos primos, usar la versión robusta para garantizar que el protocolo se ha realizado correctamente.

Una vez obtenido N se procede a calcular el valor de la clave secreta d para algún valor de la clave pública e acordado entre los participantes. En caso de que la clave pública no sea invertible el protocolo puede generar otra de manera automática. Al no disponer explícitamente de $\phi(N)$, el sistema para determinar la invalidez de dicha clave pública es probabilístico pero el nivel de consumo de recursos que requiere es pequeño en comparación con el resto del protocolo. En nuestra implementación se sigue el sub-protocolo descrito en [FMY98] sin modificaciones.

En [FMY98] la robustez se consigue eliminando a todo aquel usuario manifiestamente malicioso hasta que tan sólo queden participantes que sigan fielmente el protocolo. La detección de participantes maliciosos se consigue a través de un sistema de compromisos encadenados que parten de los fragmentos p_i y q_i y se actualizan en cada paso. Sin embargo, la fuerte interactividad y el volumen de cálculos que exigen dichos compromisos ralentiza el protocolo hasta hacerlo impracticable.

Concretamente, cada valor que aparece en el protocolo debe ser comprometido antes de ser enviado para que aquellos que lo reciban puedan verificarlo y, si procede, comprobar que tanto el valor como el compromiso son coherentes con los compromisos previos. Este proceso de generación de compromisos y verificación es sumamente costoso ya que requiere cierta sincronización entre participantes y un gran volumen de comunicaciones.

Para evitar el desperdicio de recursos que supone generar, multiplicar y chequear la primalidad de p y q de manera robusta en los casos en los que estos valores no son primos se ha optado por modificar el protocolo de la siguiente manera.

Se ejecuta la variante no-robusta del protocolo propuesta en [BF97] hasta finalizar el test de biprimalidad. Si el test da negativo los jugadores revelan los valores usados para que el resto de participantes pueda comprobar que, en efecto, $N = \sum p_i \cdot \sum q_i$ y que p o q es compuesto. Estos cálculos pueden hacerse de manera local una vez recibidos los correspondientes fragmentos de p y q. En caso de que se detecte alguna anomalía se procederá a publicar la semilla del generador pseudo-aleatorio que ha usado cada participante de manera que los demás participantes puedan verificar, de nuevo localmente, que los elementos publicados por los demás han sido correctamente calculados.

Si, por el contrario, el test da positivo, los participantes realizan la versión robusta del mismo con los mismos parámetros iniciales (p_i, q_i, semilla del generador pseudo-aleatorio, ...) para asegurarse de que todos los cálculos se han realizado correctamente sin tener que revelar ninguna información.

En caso de que en algún punto se detecte una conducta fraudulenta iniciaremos un proceso de acusación que terminará con la expulsión del protocolo de todos los participantes que hayan mentido. Para ello, resulta imprescindible que los mensajes entre participantes vayan firmados, garantizando así su integridad y el no-repudio. Acto seguido se reiniciará el protocolo sin la presencia de dichos participantes.

A pesar de que este último caso puede entorpecer el desarrollo del protocolo, ralentizándolo, no es necesario preocuparse dado que sabemos que el número de participantes maliciosos no supera el 50% de participantes del sistema (típicamente menos de 5) y cada vez que se detecta a un usuario mintiendo se lo elimina inmediatamente, evitando así que vuelva a mentir.

Además, en aquellos protocolos en los que todo comportamiento malicioso es detectado y en los que además se asegura la robustez, en el sentido de ser capaces de terminar aún en presencia de cierto número de participantes corruptos, un usuario no tiene ningún incentivo para intentar un ataque. Podemos esperar, por lo tanto, un comportamiento ideal por parte de todos los participantes.

C. Resultados obtenidos

El principal resultado obtenido por nuestra implementación ha sido hacer viable en un escenario real la generación distribuida y robusta del módulo RSA y de su clave privada. Se trata de un avance en el campo de las firmas distribuidas en procesos electorales que permite crear protocolos distribuidos y robustos sin necesidad de contar con una Tercera Parte de Confianza.

En particular, dada la capacidad de nuestra implementación para detectar y eliminar participantes maliciosos inmediatamente, la mayor parte del tiempo trabajaremos en régimen no robusto, reduciendo la complejidad de los cálculos necesarios. Además, la detección de participantes maliciosos es sencilla, pues basta con publicar los valores p_i y q_i, calcular localmente $\sum p_i \sum q_i$ y ver que coincide con el resultado obtenido en la multiplicación distribuida. Esto representa un coste añadido muy pequeño respecto al protocolo no robusto.

Por otra parte, la eliminación de usuarios tampoco supone una reducción notable de eficiencia. El número de veces que se eliminan los usuarios es muy reducido comparado con el número de iteraciones necesarias para terminar el protocolo y, para comprobar quien ha mentido, se reproduce el protocolo localmente con las semillas de generador pseudo-aleatorio del cada participante.

En resumen, con nuestra implementación se obtiene un sistema robusto y casi tan eficiente como las implementaciones del protocolo no robusto. Respecto al análisis de la complejidad del protocolo no robusto, remitimos al lector a [MWB99], donde se expone una implementación del protocolo no robusto y se analiza su eficiencia. La implementación realizada en [MWB99] consigue generar un módulo RSA de 2048 bits con 3 participantes en 18.13 minutos, un tiempo totalmente razonable.

REFERENCES

[ACS02] Joy Algesheimer, Jan Camenisch, and Victor Shoup. Efficient computation modulo a shared secret with application to the generation of shared safe-prime products. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 417–432. Springer, 2002.

[BF97] Dan Boneh and Matthew K. Franklin. Efficient generation of shared rsa keys (extended abstract). In Burton S. Kaliski Jr., editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 425–439. Springer, 1997.

[DD05] Ivan Damgård and Kasper Dupont. Efficient threshold rsa signatures with general moduli and no extra assumptions. In Serge Vaudenay, editor, *Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 346–361. Springer, 2005.

[DF02] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In Yuliang Zheng, editor, *ASIACRYPT*, volume 2501 of *Lecture Notes in Computer Science*, pages 125–142. Springer, 2002.

[DM10] Ivan Damgård and Gert Læssøe Mikkelsen. Efficient, robust and constant-round distributed rsa key generation. In Daniele Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 183–200. Springer, 2010.

[FMY98] Yair Frankel, Philip D. MacKenzie, and Moti Yung. Robust efficient distributed rsa-key generation. In *PODC*, page 320, 1998.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[MWB99] Michael Malkin, Thomas D. Wu, and Dan Boneh. Experimenting with shared generation of rsa keys. In *NDSS*. The Internet Society, 1999.

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 1991.

[RSA78] RL Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):126, 1978.

[Sch07] B. Schneier. *Applied cryptography: protocols, algorithms, and source code in C*. A1bazaar, 2007.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[Sho00] Victor Shoup. Practical threshold signatures. In Bart Preneel, editor, *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 207–220. Springer, 2000.

El proceso de Iniciativa Legislativa Popular por medio de firmas digitales

Cristina Pérez-Solà[1]
[1]Dept. Eng. Informació i Comunicacions (DEIC)
Universitat Autònoma de Barcelona
{cperez, jherrera}@deic.uab.cat

Apol·lonia Martínez Nadal[2]
[2] Facultat de Dret
Universitat de les Illes Balears
apollonia.martinez@uib.es

Jordi Herrera-Joancomartí[1,3]
[3] KISON research group
Universitat Oberta de Catalunya
Barcelona

***Resumen*—En España, la Constitución de 1978 reconoce (y la Ley Orgánica 3/1984, de 26 de marzo, regula) la denominada Iniciativa Legislativa Popular (ILP), consistente en un proceso en virtud del cual los ciudadanos pueden presentar proposiciones de ley suscritas por un número mínimo de firmantes. Con la aparición de los dispositivos de firma electrónica como el DNI electrónico, los procesos de ILP pueden llegar a sufrir una importante transformación puesto que la posibilidad de recogida de firmas digitales puede llegar a ser muy versátil. En este artículo se analiza, tanto desde el punto de vista técnico como el punto de vista jurídico, como la recogida de firmas mediante mecanismos digitales puede afectar a las ILP.**

Palabras clave: Iniciativa legislativa popular, firmas digitales, DNI electrónico.

I. Introducción

La Constitución Española, en su art. 87.3, prevé la participación directa de los ciudadanos en el proceso de producción normativa, configurando al pueblo, mediante la presentación de 500.000 firmas, como sujeto de la iniciativa legislativa. Esta participación prevista constitucionalmente se encuentra regulada por la Ley Orgánica 3/1984, reguladora de la Iniciativa Legislativa Popular (ILP), que, tras ser modificada por la Ley Orgánica 4/2006, incorpora un nuevo art. 7.4, que establece, respecto de la naturaleza de las firmas, que "Las firmas se podrán recoger también como firma electrónica conforme a lo que establezca la legislación correspondiente". La posibilidad de realizar recogidas de firmas electrónicas para ILP se confirma en la sesión de la Junta Electoral Central del 17 de septiembre de 2009. Por tanto, a la vista de lo expuesto, puede afirmarse que existe un marco legal que posibilita el desarrollo de una ILP a través de la recogida de firmas electrónicas. Cuestión distinta es la utilización que se haga en la práctica de esta posibilidad.

Los trabajos realizados hasta la fecha en este campo no son demasiados y las iniciativas que han recogido sus firmas a través de medios digitales son escasas.

Por un lado, encontramos distintas herramientas que permiten la firma de documentos de forma telemática. De entre ellas, destacan *CryptoApplet* [2], *ViaFirma* [3] y @Firma [4] que permiten la realización de firmas electrónicas en múltiples formatos así como su validación. En el marco de la administración electrónica, el proyecto Drupal X509 [5] ofrece un módulo para el gestor de contenidos Drupal que permite la creación de portales de participación ciudadana. El módulo permite la creación de formularios y su firma utilizando el DNI electrónico.

Sin embargo, estas herramientas están pensadas para la firma de distintos documentos por varios usuarios, gestionándose dichas firmas de forma separada. Nótese que en una ILP, los condicionantes son distintos, puesto que se pretende realizar la firma digital sobre un único documento, la proposición, por múltiples usuarios. Además, la gestión de todas las firmas es una gestión conjunta puesto que la validez de la ILP radica en la presentación de 500.000 firmas en su totalidad.

Desde una vertiente más académica, en [6] los autores proponen un sistema de recogida de firmas digitales enfocada a procesos de ILP. En dicho artículo, se presenta el diseño a alto nivel de la plataforma eFIC (Firma Ciudadana), una plataforma que permite la recogida de firmas. No nos consta, a día de hoy, que la plataforma propuesta en dicho artículo se haya implementado.

Finalmente, desde el punto de vista práctico, hasta la fecha únicamente el proceso de ILP pro-trasvase Tajo-Segura[1] contempla la recogida de firmas digitales. Esta ILP acepta firmas digitales que se pueden realizar a través de su portal web utilizando el DNI electrónico. La aplicación que se encarga de la gestión de las firmas digitales es una aplicación desarrollada por el departamento de informática de la Universidad de Murcia. Más allá del "Manual del firmante" que se encuentra disponible en la web de la Iniciativa, no se disponen de más detalles de la aplicación.

En lo que respecta al punto de vista jurídico, no tenemos constancia, a día de hoy, de la existencia entre la doctrina española de trabajos dedicados específicamente a la recogida de firmas electrónicas para el desarrollo de una ILP, por lo que el tema es abordado de forma incidental en trabajos generalistas y sin entrar a resolver la problemática jurídica planteada.

En este artículo se presenta una aplicación que permite la recogida y validación de firmas digitales para una ILP. La aplicación propuesta cumple con los requisitos establecidos por la ley y dicho cumplimiento se analiza tanto desde el punto de vista técnico como desde el punto de vista jurídico.

La organización del artículo es la siguiente. En la Sección II se analiza la ley que regula los procesos de ILP. La Sección III presenta las principales características de la aplicación desarrollada. En la Sección IV se presenta un análisis jurídico.

Finalmente, en el Sección V se presentan las conclusiones.

II. Breve descripción de la Ley Orgánica 3/1984

La presentación de una ILP da comienzo con la constitución de una comisión promotora, encargada de presentar el texto que será objeto de la Iniciativa. La comisión promotora deberá presentar el texto a la Mesa del Congreso, que llevará a cabo un examen de admisibilidad. Si la propuesta es admitida, se prosigue con la ILP.

El proceso de recogida de firmas se inicia en el momento en el que la Mesa del Congreso admite la proposición y lo comunica a la Junta Electoral Central (JEC) y termina con la entrega de las firmas recogidas a la JEC, para su posterior comprobación y recuento definitivo. Se requieren un mínimo de 500.000 firmas que deben ser recogidas y validadas en un plazo de nueve meses.

Una vez recogidas las firmas exigidas y entregadas a la JEC, ésta procederá a su comprobación y recuento definitivos, invalidando las firmas que no cumplan los requisitos establecidos. La JEC elevará al Congreso de los Diputados la certificación acreditativa del número de firmas válidas y procederá a destruir los pliegos que contenían las firmas.

Por último, se inicia la tramitación parlamentaria. Recibida la acreditación con el número de firmas exigido, la Mesa ordenará la publicación de la proposición, que quedará en condiciones de ser incluida en el orden del día del Pleno.

II-A. Proceso de recogida de firmas

El proceso de recogida de firmas se inicia cuando la comisión promotora presenta a la JEC los pliegos necesarios para la recogida de firmas, que deberán reproducir el texto íntegro de la proposición. La JEC, una vez aprobado el texto que contienen los pliegos, los devolverá sellados y numerados a la comisión promotora, que se encargará de la recogida de firmas.

Una vez la comisión promotora ha obtenido los pliegos, se procede a la recogida de firmas. Las firmas deberán ir acompañadas del nombre, apellidos, número de DNI y municipio en cuyas listas electorales se encuentre inscrito el elector y deberán ser autenticadas por un Notario, un Secretario Judicial, el Secretario municipal o por fedatarios especiales designados por la comisión promotora. La autenticación deberá incluir la fecha, que podrá ser colectiva, pliego por pliego.

El procedimiento de recogida de firmas debe finalizar con la entrega de las firmas recogidas a la JEC en el plazo de nueve meses desde la notificación. Las firmas deberán ir acompañadas de los certificados que acrediten la inscripción de los firmantes en el censo electoral como mayores de edad.

II-B. Identificación de los procesos tradicionales en un entorno digital

En este apartado, identificamos los documentos y procesos establecidos por la ley en un entorno digital.

El texto íntegro de la proposición es un documento electrónico con el mismo contenido. El equivalente digital de los pliegos serán unos documentos digitales análogos. A diferencia de los pliegos en papel, que contienen espacio para distintas firmas manuscritas, los pliegos en formato digital solamente contendrán la firma digital de un único ciudadano (no por dificultad tecnológica sino por simplificación conceptual). La información que contendrá un pliego digital será:

$$\textit{Pliego} = \{P, st_1, \mathcal{F}_{P,JEC}, \textit{nombre}, \textit{DNI}, \textit{municipio}\}$$

donde P es el documento con el texto íntegro de la proposición, $\mathcal{F}_{P,JEC}$ es la firma de la JEC sobre la propuesta y su sello de tiempo, st_1, *nombre* es el nombre y apellidos del ciudadano, *DNI* es el número del documento nacional de identidad, y *municipio* es el municipio en cuyas listas electorales se encuentre inscrito el ciudadano.

Nótese que el equivalente digital al sellado de los pliegos por parte de la JEC es la firma de la propuesta por parte de la JEC, $\mathcal{F}_{P,JEC}$, y su inclusión dentro del pliego. Esta firma digital va acompañada de un sello de tiempo st_1 que permitirá establecer el inicio del periodo de recogida de firmas.

La recogida de firmas manuscritas se convierte ahora en una recogida de firmas digitales. Los ciudadanos que deseen dar soporte a la iniciativa, deberán firmar digitalmente el *Pliego*, obteniendo $\mathcal{F}_{\textit{Pliego},C}$.

La contabilización de una firma digital en la ILP vendrá determinada (más allá de su correcta validación, que discutimos más adelante) por la validez legal de la firma digital emitida. Dicha validez dependerá de la autoridad de certificación que haya emitido el certificado digital que cada ciudadano esté utilizando para realizar la firma. Para ello, el sistema que realiza la validación de las firmas digitales deberá especificar las autoridades de certificación aceptadas para la firma de una propuesta. Dichas autoridades debe fijarlas la JEC.

La correcta contabilización de la firma para la propuesta de ILP debe contener las siguientes validaciones:

1. La firma debe estar realizada correctamente por el suscriptor del certificado digital.
2. El certificado digital debe estar vigente a fecha de realización de la firma.
3. La firma no se ha realizado antes de la fecha de inicio de recogida de firmas.
4. La firma no se ha realizado con posterioridad a la fecha límite de recogida de firmas.
5. Un usuario no puede firmar más de una vez.
6. El usuario debe ser mayor de edad.

Dichas validaciones las puede efectuar tanto la comisión promotora, como la JEC. En la Sección III se describen los mecanismos específicos que se han implementado en la aplicación para realizar dichas validaciones.

La entrega de las firmas digitales a la junta electoral central se realizará en soporte digital. Para ello, se remitirán todos los documentos, $Pliego$, y sus correspondientes firmas, $\mathcal{F}_{\textit{Pliego},C}$ para que la JEC pueda realizar las comprobaciones oportunas para efectuar el recuento final.

III. Aplicación de recogida de firmas

La aplicación de recogida de firmas desarrollada utiliza una arquitectura estándar cliente-servidor. El servidor dispone

de una base de datos donde se almacenan las proposiciones creadas y las firmas realizadas para cada una de ellas. También el servidor es el encargado de realizar todas las tareas de validación y sellado de tiempo. El cliente, para realizar la firma digital de la proposición deseada, sólo necesita un navegador web y un par de claves certificadas por una autoridad de certificación reconocida por la JEC. En el caso que las claves no estén en el navegador web del usuario, éste puede precisar también dispositivos para el acceso a las claves (como un lector de tarjetas inteligentes en el caso del DNI electrónico).

La aplicación no se ha desarrollado como una aplicación *standalone* sino que se ha desarrollado como un componente de Joomla, un gestor de contenidos de código abierto. Esto permite integrar la aplicación de recogida de firmas en cualquier sitio construido sobre esta plataforma. El componente se ha desarrollado utilizando PHP como lenguaje de programación y MySQL como sistema gestor de base de datos.

La aplicación dispone de tres perfiles de usuario diferenciados: el administrador, que se ocupará del funcionamiento de la plataforma; el gestor, la persona que creará las proposiciones; y el usuario, el ciudadano que podrá firmar las mismas.

El administrador es el encargado de gestionar la plataforma y, como tal, dispone de una interfaz especial donde configurar las diversas opciones. Las tres principales acciones de administración son la configuración de las autoridades de certificación y sellado de tiempo, la configuración de la visibilidad de las proposiciones y la descarga de las firmas realizadas. En primer lugar, para que un gestor pueda crear una nueva proposición que acepte una determinada autoridad de certificación, el administrador tiene que haber insertado la autoridad de certificación en la base de datos junto con su certificado. La segunda de las acciones de administración, la configuración de la visibilidad de las proposiciones, permite decidir si las propuestas son publicadas inmediatamente después de ser creadas o si, por el contrario, necesitan la aprobación del administrador antes de ser publicadas. La tercera de las acciones, la descarga de firmas, permite realizar el último trámite del procedimiento de recogida de firmas. Esta opción crea un fichero con el texto de la propuesta y las firmas efectuadas, que deberá ser entregado a la JEC para su validación y recuento final.

El gestor es la persona que crea la proposición en la plataforma y ejerce, por tanto, el papel de comisión promotora. En el momento que el gestor crea la proposición, éste establece cuales de las autoridades de certificación que el administrador ha configurado desea que sean reconocidas para firmar la proposición. El hecho de aceptar unas autoridades de certificación u otras permitirá crear proposiciones válidas para ser presentadas como ILP o recogidas de firmas más informales.

Los usuarios de la plataforma son los ciudadanos que desean firmar las proposiciones. Éstos pueden listar las proposiciones creadas (que han sido aprobadas por el administrador) y firmar las que deseen. Una vez localizada la proposición a firmar, el usuario puede acceder al contenido completo de la misma así como a la opción de firma. La opción de firma se encuentra disponible únicamente en la página que contiene el texto íntegro de la proposición, asegurando que los usuarios tengan conocimiento de la iniciativa que suscriben. Además, el propio navegador exige una confirmación por parte del usuario antes de realizar la firma. En algunos navegadores, el mismo diálogo de confirmación muestra el texto íntegro a firmar, reafirmando el conocimiento del texto que se está firmando. Para que un usuario pueda firmar una proposición, es necesario que éste disponga de un certificado de firma emitido por alguna de las autoridades de certificación que fueron seleccionadas por el gestor en el momento de la creación. En caso que así sea, el usuario puede proceder a realizar la firma de la proposición.

III-A. Validaciones que realiza la aplicación

Más allá de las funcionalidades de gestión que realiza la aplicación, un punto muy importante son las validaciones que ésta realiza sobre las firmas digitales que obtiene, puesto que la eficacia de la aplicación radica en la verificabilidad de las firmas obtenidas. En la aplicación se han implementado aquellas validaciones que realizará posteriormente la JEC para certificar el número de firmas válidas.

En primer lugar, se comprueba que la firma haya sido realizada correctamente por el suscriptor del certificado digital. Esta comprobación permite asegurar que la firma realizada por el suscriptor corresponda a una firma del pliego en cuestión. Esta verificación es necesaria puesto que la firma digital se realiza en el dispositivo del cliente y el texto que se firma puede haber sido modificado y puede diferir, por tanto, del texto original de la proposición. Para que una firma sea considerada como válida también es necesario comprobar que el certificado con el cual se ha realizado es adecuado.

En segundo lugar, se verifica que el certificado que el ciudadano ha usado para firmar está vigente a fecha de realización de la firma. Esta comprobación requiere la verificación del periodo de vigencia del certificado y de su estado de revocación. La verificación del periodo de vigencia se realiza comprobando los campos de validez del certificado. La comprobación del estado de revocación del certificado es llevada a cabo a través del protocolo OCSP. Esta validación precisa de un sello de tiempo que se puede incluir en cada una de las firmas digitales y que deberá ser realizado por una autoridad de sellado de tiempo reconocida por la JEC. Si bien esta opción es técnicamente válida, la obtención de un sellado de tiempo tiene un coste económico asociado (del orden de 20.000 euros para 500.000 sellos), de modo que el sellado de tiempo de cada una de las firmas recogidas incrementa de forma muy notable el coste económico de la ILP. Sin embargo, dado que la ley permite que el establecimiento de la fecha puede ser colectivo, se puede aplicar a un conjunto determinado de firmas digitales un único sellado de tiempo, acreditando de este modo que dichas firmas no se han realizado con posterioridad a la fecha del sellado. Es importante destacar que, la opción del sellado de tiempo colectivo implica que la fecha de realización de la firma difiera de la fecha en la que se puede demostrar que efectivamente se ha realizado la firma (fecha en la que se realiza el sellado de tiempo) y por tanto es necesario que el certificado sea válido en esta segunda, de

modo que, de forma efectiva, la comprobación de la vigencia del certificado es realizada después del sellado de tiempo.

En tercer lugar, es necesario comprobar que la firma no se haya realizado antes de la fecha de inicio del periodo de recogida de firmas. Nótese que la firma $\mathcal{F}_{Pliego}$ ya asegura que la firma no se ha realizado antes de la fecha de inicio de recogida de firmas puesto que el ciudadano incluye en su firma el valor $\mathcal{F}_{P,JEC}$ que equivale al sellado del pliego por parte de la JEC y que ya incluye la fecha en la que la JEC realizó la firma.

En cuarto lugar, se debe comprobar que la firma no se ha realizado con posterioridad a la fecha límite de recogida de firmas. Para asegurar este punto se dispone del sello de tiempo que se realiza para la firma, ya sea éste para cada firma o para un conjunto de ellas.

En quinto lugar, se debe asegurar que el usuario no pueda firmar más de una vez la misma proposición. El control de duplicados es llevado a cabo por el sistema gestor de la base de datos en base al número de DNI del firmante, que se extrae del certificado que éste ha usado para firmar.

Por último, queda por comprobar que el usuario que ha realizado la firma es mayor de edad. Esta comprobación se puede realizar únicamente si el certificado usado para firmar contiene la fecha de nacimiento de su propietario, de lo contrario esta comprobación quedará en manos de la JEC.

III-B. Especificación del formato de los datos

Las firmas realizadas por los usuarios son almacenadas en una base de datos relacional. En concreto, se almacena un objeto signedData de PKCS#7 [7] codificado en base 64, que contiene tanto el certificado del usuario como el de la autoridad de certificación que lo ha emitido y la firma realizada. El objeto PKCS#7 no contiene el texto que se ha firmado, es decir, la firma es *detached*. Esto supone un ahorro de espacio considerable puesto que, en caso contrario, sería necesario almacenar el texto a firmar tantas veces como firmas se hubieran realizado. De este modo, el espacio estimado para almacenar las 500.000 firmas de una ILP es de 2GB que pueden ser incrementados hasta 6GB si se añaden sellos de tiempo individuales para cada firma y se almacenan las respuestas de las comprobaciones de revocación de certificados.

Tanto las comprobaciones de revocación de certificados como los sellos de tiempo realizados sobre las firmas son almacenados en la base de datos, los primeros codificados en base 64 y los segundos en el formato de *timestamp response* especificado en la RFC 3161 [8] codificado en base 64.

Una vez se ha alcanzado el número de firmas necesarias, éstas deben ser presentadas ante la JEC. La exportación de las firmas se realiza mediante un fichero XML que contiene toda la información relativa a la proposición. El documento contiene el texto completo de la proposición, todas las firmas recogidas con sus respectivas comprobaciones de revocación y los sellos de tiempo realizados sobre las firmas.

IV. Análisis jurídico

Como se ha expuesto, tanto Ley Orgánica 3/1984, de 26 de marzo, reguladora de la Iniciativa Legislativa Popular (tras la reforma introducida por la Ley Orgánica 4/2006, de 26 de mayo) permite declarar, de forma genérica, la admisibilidad legal de una recogida de firmas realizada por medios electrónicos. A partir de esta admisibilidad genérica, se trata de analizar el cumplimiento de los distintos requisitos que de forma concreta se establecen legalmente para el desarrollo de una ILP en caso de que esta se desarrolle por medios electrónicos; y en concreto, el cumplimento de esos requisitos por parte de la aplicación informática objeto de este trabajo.

1. El procedimiento de recogida de firmas. Conforme al art. 7 LOILP, admitida la proposición puede iniciarse el procedimiento de recogida de firmas. Y la admisión genérica de firmas digitales nos plantea distintas cuestiones y dudas a la vista de la concreta regulación del procedimiento de recogida de firmas establecida en los siguientes preceptos:

A. Requisitos formales previos: los pliegos para la recogida de firmas (art. 8). Conforme a la LOILP (art.8), y para el supuesto de recogida de firmas manuscritas tradicionales, la primera actuación de la Comisión Promotora, una vez admitida la proposición, es presentar ante la JEC los pliegos necesarios, en papel de oficio, para la recogida de las firmas.

Para el supuesto de firmas digitales, partimos del presupuesto de que la finalidad de tales exigencias de pliegos debidamente sellados y numerados es evitar posibles manipulaciones, falsedades o confusiones. Y, en la aplicación objeto de estudio entendemos que, efectivamente, se consiguen similares garantías por cuanto, de entrada, el texto íntegro de la propuesta legislativa es firmado por la JEC que, además, incluirá el correspondiente sello temporal. Esta firma, junto con su inclusión, como veremos, en el pliego digital, es el equivalente funcional del sellado del art.8; y con esta firma se evita cualquier manipulación del contenido de la propuesta. Además, en la aplicación objeto de estudio, el equivalente digital de los pliegos son unos documentos digitales análogos, unos "pliegos digitales" que contendrán, entre otra información, el documento con el texto íntegro de la proposición, la firma de la JEC sobre la propuesta y su sello de tiempo (en cumplimiento de las exigencias del art. 8 LOILP), y también el nombre y apellidos del ciudadano, el número del documento nacional de identidad, y el municipio en cuyas listas electorales se encuentre inscrito el ciudadano (en cumplimiento de las exigencias del art. 8 LOILP). Y cada uno de estos pliegos contendrá la firma digital de un ciudadano, firma que afectará a la totalidad de la información contenida en el pliego digital. De esta forma, los pliegos que contienen el texto íntegro de la propuesta firmada por la JEC quedan vinculados a la firma de los ciudadanos, por cuanto el documento que firman los usuarios es, precisamente, la propuesta firmada por la JEC; de esta forma se evitan posibles actuaciones fraudulentas (en el sentido de incluir firmas suscritas con finalidades distintas o utilizar las firmas recogidas para otros objetivos).

Por otra parte, la exigencia del art. 8 de que los pliegos reproduzcan el texto íntegro de la proposición, entendemos persigue la finalidad de que los firmantes tengan conocimiento pleno y fundada de la iniciativa que suscriben; y entendemos

que ello puede conseguirse igualmente, e incluso de forma más plena y eficaz, en el caso de ILP telemática incluyendo en la página de recogida de datos un enlace a la propuesta o incluso, configurándolo como página de paso obligatorio.

Además, recuérdese que, en la aplicación objeto de estudio, esta firma digital de la Junta Electoral va acompañada de un sello de tiempo de especial relevancia a efectos de determinar la validez temporal de las firmas recogidas. En efecto, como es sabido, sólo las firmas recogidas dentro de plazo son firmas válidas, no siendo admisibles las recogidas con anterioridad al inicio del periodo de recogida, ni las suscritas con posterioridad a la finalización de dicho plazo. De ahí la importancia de la prueba del momento temporal de realización de la firma. En la aplicación objeto de análisis, en caso de descartar, por sus costes, el sellado temporal individual de cada una de las firmas, se proponen otros medios indirectos de prueba de la validez temporal de la firma recogida que pueden considerarse igualmente válidos, por cuanto aun cuando no prueban el momento exacto en que se ha realizado tal firma si consiguen probar su realización dentro del periodo legalmente válido de recogida.

B. Momento de la firma. Los ciudadanos que deseen apoyar la iniciativa legislativa propuesta deberán firmar digitalmente el pliego digital. Y esta actuación nos plantea, jurídicamente, dos cuestiones.

Los firmantes: sujetos legitimados para firmar. De acuerdo con lo dispuesto en la LOILP (art. 1), pueden ejercer la ILP los ciudadanos españoles mayores de edad que se encuentren inscritos en el censo electoral (y en pleno ejercicio del derecho de sufragio activo).

Y, precisamente, a efectos de identificación y comprobación de la legitimación de los firmantes, el art. 9 LOILP, relativo a la autenticación de las firmas, dispone que, junto a la firma del elector se indicará su nombre y apellidos, número del documento nacional de identidad y municipio en cuyas listas electorales se halle inscrito.

En el supuesto de recogida de firmas digitales, entendemos que la identificación y comprobación de la legitimación puede realizarse con igual o incluso superior seguridad y, sin ninguna duda, con mayor rapidez y eficacia. En concreto, en el caso de la aplicación objeto de estudio, si la firma digital admitida es la incorporada al denominado DNI electrónico, será posible extraer esos datos personales de la misma información incorporada al chip electrónico del mismo y la aplicación puede realizar la comprobación de la mayoría de edad.

La firma digital; clases de firma admisibles. Dado lo novedoso del sistema de firma digital que admite el art. 7.4 LOILP, la escasa doctrina que ha abordado el tema considera necesario esperar a una norma de desarrollo que establezca los requisitos de seguridad que hagan posible y fiable el recurso a la firma electrónica para la recogida de firmas. Y doctrinalmente, se han apuntado dos posibilidades respecto del impulso de esta norma de desarrollo: un acuerdo de la Mesa del Congreso de los Diputados o una Instrucción o Resolución de la JEC, considerándose más correcta esta última, por cuanto, conforme al art. 7 LOILP, corresponde a la JEC el control de la regularidad del procedimiento de recogida de firmas [9].

En la práctica, de forma reciente, se ha optado por una vía próxima a esta segunda solución. En concreto, hemos de referirnos al Acuerdo de la Junta Electoral de 28 de mayo de 2009 que supone su primer pronunciamiento respecto del uso de la firma digital en el desarrollo de ILP y que dispone que la Comisión promotora debe comunicar a la JEC "el sistema de firma electrónica que pretenda utilizar y facilitar a ésta, en el caso de que fuera necesario, el sistema utilizado para la verificación de las firmas electrónicas". La finalidad de tal comunicación entendemos que es la validación por parte de la Junta de la validez legal del sistema de firma. Obsérvese, pues, que la Junta Electoral, tras reiterar la admisibilidad de la recogida de firmas digitales para los procedimientos de ILP no establece a continuación los requisitos que de forma general deban cumplir los sistemas de firma digital (ni siquiera remite a una resolución o instrucción de desarrollo posterior) sino que adopta el mecanismo de la comunicación previa y autorización para cada caso concreto.

Y, por ello, este primer acuerdo va seguido de otro Acuerdo de 28 de enero de 2010 que nos sitúa ya ante la comunicación previa de un concreto sistema de firma digital por parte de la Comisión Promotora (la Comisión Promotora de la ILP Tajo-Segura) que pretende utilizar las nuevas tecnologías para la recogida de firmas y pretende obtener la autorización a tal efecto de la JEC. Y tal autorización se otorga por la mencionada Junta con el informe previo de la Oficina del Censo Electoral (que, conforme a la LOILP, interviene en la comprobación y recuento previos de las firmas recogidas).

Vistos estos aspectos procesales relativos a la autorización del sistema de firma, nos centramos ahora en aspectos sustantivos relativos a la clase de firma admisible. Y, en principio, ha de tenerse en cuenta la regulación de la firma electrónica que nos sitúa en la Ley 59/2003, de 19 de diciembre, de firma electrónica; y, de entre las distintas clases de firma electrónica que contempla la Ley, nos inclinamos por la utilización, de la denominada firma electrónica reconocida que, por sus características, está equiparada legalmente a la firma manuscrita (en virtud del art. 3 de la Ley 59/2003). Mención especial merece el denominado DNI electrónico, regulado inicialmente en los art. 15 y 16 de la Ley 59/2003 y en el Real Decreto 1553/2005, de 23 de diciembre (cuyo art. 1.5 lo equipara también a la firma manuscrita). Por ello, dada la seguridad que ofrece la firma incorporada, y dado el grado de difusión actual del DNI electrónico, resulta un instrumento especialmente adecuado para la recogida de firmas digitales para el desarrollo de una ILP.

C. Autenticación de la firma. A fin de evitar fraudes, los art. 9 y 10 LOILP establecen las exigencias de autenticación de las firmas, exigencias cuyo cumplimiento constituye, como hemos visto, una de las condiciones impuestas por la Junta Electoral para la admisibilidad de las firmas digitales.

En concreto, dispone el art. 9.2 que, en principio, la firma debe ser autenticada por un Notario, por un Secretario Judicial o por el Secretario municipal correspondiente al municipio en cuyo

censo electoral se halle inscrito el firmante.

Junto a esta autenticación realizada por los fedatarios tradicionales, el art. 10 de la LOILP abre una interesante posibilidad para los supuestos de ILP electrónica, en la medida que establecen que "las firmas podrán también ser autenticadas por fedatarios especiales designados por la Comisión Promotora". Posibilidad que permite que actúen como fedatarios personas con los conocimientos técnicos suficientes y necesarios para la autenticación de firmas electrónicas, personas que podrían ser propuestas por el administrador de la aplicación informática de entre sus empleados, lo que permitiría un procedimiento de autenticación más ágil y eficaz. Téngase en cuenta que para adquirir esta condición de fedatarios especiales debemos estar ante "ciudadanos españoles que, en plena posesión de sus derechos civiles y políticos y careciendo de antecedentes penales, juren o prometan ante las Juntas Electorales provinciales dar fe de la autenticidad de las firmas de los signatarios de la proposición de Ley" (art. 10.2); y que, dada la transcedencia de su actuación, la adquisición de tal condición no es irrelevante pues, en caso de falsedad, los fedatarios especiales puede incurrir en responsabilidad penal (art. 10.3).

En cualquier caso, cabe plantear en qué consistiría la autenticación en el supuesto de recogida de firmas digitales. En este caso, partimos de la utilización de certificados identificativos que permiten articular la presunción iuris tantum de que quien está firmando es efectivamente el titular del par de claves certificadas; por ello, la autenticación o bien sería inexistente, por innecesaria, o bien consistiría en la validación o verificación de la firma electrónica, en el sentido de comprobar que se cumplen con los requisitos de verificación, técnicos y jurídicos, necesarios para considerarla una firma válida, requisitos ya expuestos en la parte técnica de este trabajo. Y obsérvese que estas mismas actuaciones son las que deberían efectuar nuevamente tanto la Oficina del Censo Electoral como la JEC a efectos de la comprobación inicial y el recuento definitivo de firmas.

2. Remisión de los pliegos a la Junta Electoral. Comprobación y recuento de firmas. Una vez finalizada la recogida de firmas y su autenticación, conforme al art. 11.1 LOILP los pliegos que contengan las firmas se han de enviar a la JEC. La JEC, a su vez, los remite a la Oficina del Censo Electoral para que acredite la inscripción de los firmantes en el Censo Electoral como mayores de edad, y lleve a cabo la comprobación y el recuento inicial de dichas firmas. La Oficina del Censo Electoral, en el plazo de quince días, remitirá a la Junta Electoral Central certificación de todo ello (art. 11). Y, conforme al art. 12.1, una vez remitidos los pliegos a la JEC, esta procederá a su comprobación y recuento definitivos. En este momento, las "firmas que no reúnan los requisitos exigidos en esta Ley se declararán inválidas y no serán computadas" (art. 12.2).

En la aplicación objeto de estudio, la entrega de las firmas digitales a la junta electoral central se realizará en soporte digital; para ello, se remitirán todos los documentos o pliegos y sus correspondientes firmas, para que la JEC pueda realizar las comprobaciones oportunas para efectuar el recuento final. Recuérdese asimismo que en el Acuerdo de 28 de mayo de 2010 de la Junta Electoral se establece que la Comisión promotora, además de comunicar a la Junta Electoral Central el sistema de firma electrónica que pretenda utilizar, debe también "facilitar a ésta, en el caso de que fuera necesario, el sistema utilizado para la verificación de las firmas electrónicas". Comprobado el cumplimiento de los requisitos exigidos para la válida presentación de la proposición (en concreto, la consecución de las, como mínimo, 500.000 firmas válidas), la JEC elevará al Congreso de los Diputados certificación acreditativa del número de firmas válidas y procederá a destruir los pliegos de firmas que obren en su poder.

Y, a partir de ese momento se iniciaría la tramitación parlamentaria de la proposición, conforme al art. 13 LOILP, lo que no supone, en modo alguno la aprobación de la proposición, que queda en manos del Pleno del Congreso.

V. Conclusiones

En este trabajo hemos presentado de forma breve una aplicación que permite la recogida de firmas digitales para la realización de una iniciativa legislativa popular. Hemos descrito sus principales características y hemos detallado las validaciones que se llevan a cabo para cumplir con los requisitos legales que marca la ley. Por otro lado, hemos realizado un análisis jurídico del proceso de recogida de firmas digitales que nos permite concluir que aun son mejorables las normativas al respecto, por cuanto sigue sin existir una regulación detallada de los requisitos que deben cumplirse para la recogida de firmas digitales.

Agradecimientos

Este trabajo está parcialmente financiado por el Ministerio de Ciencia y Educación, a través de los proyectos *CONSOLIDER CSD2007-00004* y *TSI2006-03481* así como por el proyecto AVANZA (TSI-020501-2008 -191).

Referencias

[1] ILP en defensa del trasvase Tajo-Segura, "Defendemos el trasvase", http://www.defendemoseltrasvase.es/index.php

[2] Paúl Santapau Nebot, Ricardo Borillo Domenech, "Cryptoapplet. Applet de realización de firma digital multiformato", Universitat Jaume I.

[3] Grupo Viavansi, "Viafirma", http://www.viafirma.com/.

[4] Ministerio de Administraciones Públicas, "@firma. plataforma de validación y firma electrónica",

[5] José Luis Blasco Díaz, Modesto Fabra Valls, Manuel Mollar Villanueva, Paúl Santapau Nebot, and Santiago Manzano Romero, "Drupal aplicado a la administración electrónica".

[6] R. Conde Melguizo, A. Muñoz Muñoz, and M. L. González González, "Recogida de firmas electrónicas dentro del marco legal español. eparticipación basada en edni.", in *CollECTeR Iberoamérica*, 2008.

[7] B. Kaliski, "PKCS 7: Cryptographic Message Syntax Version 1.5", RFC 2315 (Informational), Mar. 1998.

[8] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161 (Proposed Standard), Aug. 2001.

[9] E. Aranda Alvarez, "La nueva Ley de Iniciativa Legislativa Popular", Revista Española de Derecho Constitucional, n. 76, 2006, pág. 187-218.

[10] J. Marco Marco, "El letargo participativo: la iniciativa legislativa popular en España", Revista de las Cortes Generales, n. 69, 2006, pág. 39-82.

Un Criterio de Privacidad Basado en Teoría de la Información para la Generación de Consultas Falsas

David Rebollo-Monedero
Dpto. Ingeniería Telemática,
Universitat Politècnica de Catalunya
david.rebollo@entel.upc.edu

Javier Parra-Arnau
Dpto. Ingeniería Telemática,
Universitat Politècnica de Catalunya
javier.parra@entel.upc.edu

Jordi Forné
Dpto. Ingeniería Telemática,
Universitat Politècnica de Catalunya
jforne@entel.upc.edu

***Resumen*—En este artículo presentamos un criterio de privacidad basado en teoría de la información para la generación de consultas falsas en el ámbito de la recuperación de información privada. Medimos el riesgo de privacidad como la divergencia de Kullback y Leibler entre la distribución de consultas del usuario y la de la población, que incluye la entropía de la distribución del usuario como caso especial. Asimismo, llevamos a cabo una rigurosa justificación de nuestra métrica al interpretarla desde distintas perspectivas de teoría de la información, desde la propiedad de equipartición asintótica, pasando por los fundamentos sobre los que sustentan los métodos de maximización de la entropía, la minimización de la divergencia y la minimización de la ganancia de información, hasta el lema de Stein.**

I. Introducción

Durante las últimas dos décadas, Internet se ha ido integrando de manera gradual en nuestra vida diaria. Una de las actividades más frecuentes que llevan a cabo los usuarios cuando navegan por la Web es enviar una consulta a un motor de búsqueda. Los motores de búsqueda permiten a los usuarios recuperar información sobre una gran variedad de categorías, tales como hobbies, deportes, negocios o salud. Sin embargo, la mayoría de usuarios no son conscientes de los riesgos de privacidad que ello entraña [1].

De noviembre a diciembre de 2008, el 61 % de los adultos en Estados Unidos buscaron información en la red sobre una enfermedad en particular, un tratamiento específico, y otros temas relacionados [2]. Dichas consultas podrían revelar información sensible y ser utilizada para construir perfiles de usuario sobre enfermedades potenciales. Esta información privada podría acabar más tarde en las manos de un empresario y frustrar las esperanzas de uno de sus empleados.

En la literatura sobre sistemas de recuperación de información abundan los casos como el descrito, en los que se constata la importancia de la privacidad del usuario. Estos casos incluyen no sólo el riesgo de que los usuarios puedan ser caracterizados por un motor de búsquedas de Internet, sino también por proveedores de servicios basados en la localización (LBS, *location-based services*), o incluso la caracterización de empresas por parte de proveedores de bases de datos de patentes o mercados de valores. En este contexto, la falsificación de consultas, que consiste en acompañar consultas auténticas con consultas falsas, emerge como una posible solución para garantizar la privacidad del usuario hasta un cierto punto, a costa de una sobrecarga de tráfico y procesado.

Este artículo presenta un nuevo criterio de privacidad basado en teoría de la información para la generación de consultas en el ámbito de la recuperación de información. En concreto, nuestro criterio mide el riesgo de privacidad como una divergencia entre la distribución de consultas del usuario y la de la población, y contempla la entropía de la distribución del usuario como un caso particular. El objeto de este artículo es interpretar y justificar nuestra métrica de privacidad desde distintas perspectivas, a través de la propiedad de equipartición asintótica, el test de hipótesis y el lema de Stein.

La Sección II revisa las propuestas más relevantes en cuanto a recuperación de información privada y criterios de privacidad. La Sección III repasa algunos conceptos fundamentales relacionados con teoría de la información que ayudarán a entender la esencia de este trabajo. La Sección IV presenta una formulación de teoría de la información sobre el compromiso entre privacidad y redundancia para la falsificación de consultas en el contexto de recuperación de información privada. Esta sección muestra nuestra medida de privacidad, y posteriormente la interpreta y justifica. Finalmente, en la Sección V se presentan las conclusiones.

II. Estado del Arte en Recuperación de Información Privada

A lo largo de este artículo, utilizaremos el término recuperación de información privada (PIR, *private information retrieval*) en su sentido más amplio, queriendo decir con ello que no nos ceñiremos a las técnicas basadas en criptografía normalmente relacionadas con este acrónimo. Por consiguiente, nos referiremos a un escenario más genérico en el que los usuarios envían consultas de propósito general a un proveedor de servicios de información. Un ejemplo sería un usuario que enviase la consulta: "¿Cuál es la película más taquillera en la categoría de ciencia ficción?". A continuación, revisaremos las contribuciones más destacadas para PIR sobre la generación de consultas falsas y criterios de privacidad.

II-A. Recuperación de Información Privada

En el ámbito de la recuperación de información privada, existen una gran variedad de propuestas. Algunas de ellas se basan en terceras partes de confianza (TTPs, *trusted third parties*) que actúan como intermediario entre los usuarios y el proveedor de servicios de información [3]. Aunque este tipo

de soluciones garantizan la privacidad del usuario gracias a que su identidad es, de hecho, desconocida para el proveedor de servicios, la confianza del usuario únicamente se traslada de una entidad a otra.

Como alternativa, algunas propuestas que no se basan en TTPs, utilizan técnicas de perturbación. En el caso concreto de LBS, los usuarios perturban su información de localización al consultar a un proveedor de servicios [4]. Esto proporciona a los usuarios un cierto nivel de privacidad en términos de localización, pero no así en cuanto al contenido de las consultas y la actividad del usuario. Asimismo, esta técnica plantea un compromiso entre privacidad y utilidad de los datos: cuanto mayor es la perturbación de la localización, mayor es la privacidad del usuario, pero menor la precisión de las respuestas del proveedor de servicios. Como alternativa, los métodos criptográficos para PIR permiten a un usuario recuperar, de forma privada, el contenido de una base datos indexado por una dirección de memoria enviada por el usuario, haciendo que sea inviable por parte del proveedor de la base de datos averiguar qué entradas fueran recuperadas [5]. Desafortunadamente, este tipo de métodos requieren la cooperación del proveedor en el protocolo de privacidad, se restringen hasta cierto punto a funciones de consulta-respuesta en forma de tablas de búsqueda de longitud finita con respuestas precomputadas, y conllevan una significativa carga computacional.

La generación de consultas falsas, que el centro de nuestra discusión, aparece como una alternativa a los métodos anteriores. La idea subyacente consiste en enviar consultas originales junto con consultas falsas. A pesar de la sencillez de este método, la falsificación de consultas es capaz de garantizar la privacidad del usuario hasta un cierto punto, a costa de una sobrecarga de tráfico y procesado, aunque sin tener que tener confiar ni en el proveedor de información ni en el operador de red.

Basándose en este principio, se han propuesto e implementado varios protocolos PIR. En [6], [7], se presenta una solución que pretende preservar la privacidad de un grupo de usuarios que navegan por Internet compartiendo un punto de acceso a la Web. Los autores proponen la generación de transacciones falsas, i.e., accesos a páginas web para frustrar a un atacante en su intento por caracterizar al grupo. La privacidad se mide como la similitud entre el perfil real de un grupo de usuarios y el observado por el atacante [6].

Además de las implicaciones legales, existen distintas consideraciones técnicas para la preservación de la privacidad a través de la generación de consultas falsas [8], puesto que los atacantes podrían analizar no sólo el contenido de las consultas sino también la actividad, el ritmo de generación, el enrutamiento o cualquier otro parámetro del protocolo de transmisión, por medio de varias consultas o a través de diversos servicios de información. Asimismo, se espera que tanto los proveedores de información como los de la red se muestren reticentes a la generación automática de consultas falsas, con lo que cualquier esquema que se precie debe tener en cuenta la sobrecarga de tráfico.

II-B. Criterios de Privacidad

En esta sección revisaremos una serie de técnicas propuestas originalmente para el control de revelación estadístico (SDC, *statistical disclosure control*), pero igualmente aplicables a PIR, la aplicación que motiva nuestro trabajo. En privacidad de bases de datos, se define un *conjunto de microdatos* como una tabla de base de datos cuyos registros contienen información sobre encuestados individuales. Específicamente, este conjunto contiene atributos clave, es decir, atributos que, utilizados conjuntamente, se pueden relacionar con información externa para reidentificar a los encuestados a los que se refieren los registros en el conjunto de microdatos. Como ejemplo, los atributos clave podrían ser trabajo, dirección, edad, género, peso y altura. De igual modo, el conjunto de microdatos contiene atributos confidenciales con información sensible sobre el encuestado, tales como sueldo, religión o afiliación política.

Un planteamiento habitual en SDC es la microagregación, que consiste, primero, en dividir el conjunto de datos en grupos de registros con tuplas de valores de atributos clave similares, y segundo, en reemplazar las tuplas de cada registro en cada uno de los grupos por una tupla representativa del grupo. Uno de los criterios de privacidad más populares en la anonimización de bases de datos, es k-anonimato [9]. Este criterio se puede lograr a través de la microagregación, ya que requiere que cada combinación de atributos clave sea compartida por al menos k registros en el conjunto de microdatos. Sin embargo, el principal inconveniente de este criterio y de sus posteriores mejoras [10]–[12] es su vulnerabilidad ante los ataques de similitud y *skewness* [13]. Con el objeto de superar estas deficiencias, [14] propone otro criterio de privacidad. Concretamente, un conjunto de datos satisface t-closeness si, para cada grupo de registros que comparten una combinación de atributos clave, la divergencia de Kullback y Leibler (KL) entre la distribución de atributos confidenciales dentro de un grupo y la distribución de estos atributos en el conjunto de datos global no supera un umbral t. Inspirados en esta idea, [15], [16] definen riesgo de privacidad como una versión promediada del requisito impuesto por t-closeness sobre el conjunto de grupos agregados. Otro criterio de privacidad basado en teoría de la información propone medir el grado de anonimato observable por un atacante como la entropía de la distribución de probabilidad de los posibles emisores de un determinado mensaje [17], [18].

A pesar de las propuestas citadas anteriormente, querríamos poner énfasis en la posible necesidad, por parte de algunas aplicaciones, de criterios de privacidad basados en teoría de la información más sofisticados que k-anonimato o sus respectivas mejoras.

III. Introducción a Conceptos de Teoría de la Información

A lo largo de este artículo, denominaremos alfabeto al espacio medible en el que una variable aleatoria (v.a.) toma valores. Seguiremos la convención de utilizar mayúsculas para las v.a.'s, y minúsculas para los valores particulares que éstas

pueden tomar. Las funciones de densidad de probabilidad (PDFs, *probability density functions*) y las funciones de masa de probabilidad (PMFs, *probability mass functions*) son denotadas por p, subindexadas por sus correspondientes v.a.'s en caso de ambigüedad. Por ejemplo, tanto $p_X(x)$ como $p(x)$ indican el valor de la función p_X en x, lo que ayuda a escribir ecuaciones más concisas. De manera informal, nos referiremos ocasionalmente a la función p como $p(x)$. Asimismo, utilizaremos la notación $p_{X|Y}$ y $p(x|y)$ de manera equivalente.

En este artículo, adoptamos la misma notación utilizada en [19] para cantidades de teoría de la información. En concreto, el símbolo H se referirá a la entropía y D a la entropía relativa o divergencia KL. A continuación recordamos muy brevemente varios conceptos de teoría de la información para aquellos lectores que no estén íntimamente familiarizados con este campo. Por simplicidad, utilizaremos logaritmos neperianos.

- La *entropía* $\mathrm{H}(X)$ de una v.a. discreta X con distribución de probabilidad p es una medida de su incertidumbre, y se define como

$$\mathrm{H}(X) = -\,\mathrm{E}\,\ln p(X) = -\sum_x p(x)\ln p(x),$$

  donde E es el operador esperanza. Este operador es sustituido por la integral cuando p es una PDF.

- Dadas dos distribuciones de probabilidad $p(x)$ y $q(x)$ sobre el mismo alfabeto, la *divergencia KL* o *entropía relativa* $\mathrm{D}(p\,\|\,q)$ se define, en el caso discreto, como

$$\mathrm{D}(p\,\|\,q) = \mathrm{E}_p\,\ln\frac{p(X)}{q(X)} = \sum_x p(x)\ln\frac{p(x)}{q(x)}.$$

  Cuando p y q son PDFs, la esperanza se transforma en una integral.

Aunque la divergencia KL no satisface la propiedad de simetría y la desigualdad triangular, nos da una medida de la distancia o discrepancia entre distribuciones, en el sentido que $\mathrm{D}(p\,\|\,q) \geq 0$, con igualdad si y sólo si $p = q$.

Este intuitivo sentido de distancia se hace más evidente al examinar el lema de Stein. Suponga que observamos una secuencia de k v.a.'s independientes e idénticamente distribuidas (i.i.d.'s), y que necesitamos evaluar si éstas han sido generadas según una distribución de probabilidad p_1, hipótesis $\mathscr{H}_1$, o p_2, hipótesis $\mathscr{H}_2$. Dadas estas dos hipótesis, definimos la *región de aceptación* $\mathscr{A}_k$ como el conjunto de secuencias que, una vez observadas, nos llevan a aceptar $\mathscr{H}_1$. De forma análoga, definimos el complemento de este conjunto, $\bar{\mathscr{A}_k}$, como el conjunto de secuencias que nos decantan por $\mathscr{H}_2$. A continuación, contemplamos las siguientes probabilidades de error:

- la probabilidad de un falso negativo α_k, definido como la probabilidad de aceptar $\mathscr{H}_2$ cuando $\mathscr{H}_1$ es cierta,
- y la probabilidad de un falso positivo β_k, definido como la probabilidad de aceptar $\mathscr{H}_1$ cuando $\mathscr{H}_2$ es cierta.

Suponga que elegimos una región de aceptación con la intención de minimizar β_k, mientras que no permitimos que α_k exceda un cierto umbral ϵ. En términos generales, el lema de Stein afirma que la tasa de error óptima, β_k^ϵ, es aproximadamente $e^{-k\,\mathrm{D}(p_1\,\|\,p_2)}$, para valores de k grandes y ϵ pequeños.

A modo de ejemplo, considere el test de hipótesis en el que observamos una secuencia $X_1, \ldots, X_k$ de k lanzamientos i.i.d.'s de una v.a., e intentamos averiguar si se han producido de acuerdo con una distribución gaussiana $p_1 = \mathscr{N}(d/2, \sigma^2)$ o $p_2 = \mathscr{N}(-d/2, \sigma^2)$. Teniendo en cuenta estas distribuciones, elegiríamos la región de decisión óptima $\mathscr{A}_k$ dada por el lema de Neyman-Pearson [19], y calcularíamos la probabilidad de un falso positivo como la integral de $p_2(x_1, \ldots, x_k)$ sobre $\mathscr{A}_k$. Resulta que, a partir del lema de Stein, esta probabilidad es aproximadamente $e^{\frac{-k\,d^2}{2\sigma^2}}$, ya que $\mathrm{D}(p_1\,\|\,p_2) = \frac{d^2}{2\sigma^2}$, lo que hace más palpable esta noción de distancia: cuanto mayor es la distancia real d entre las medias de las dos distribuciones, mayor es la divergencia KL, y menor la probabilidad de error al distinguir entre ambas distribuciones.

IV. Un Criterio de Privacidad de Teoría de la Información para la Falsificación de Consultas

Esta sección presenta la principal contribución de este trabajo, un nuevo criterio de privacidad basado en una cantidad de teoría de la información para la falsificación de consultas en PIR. En concreto, la Sección IV-A introduce nuestra medida de privacidad, lo que nos conduce al problema de optimización mostrado en la Sección IV-B en el que se presenta el compromiso óptimo entre riesgo de privacidad y redundancia. Posteriormente, interpretamos y justificamos nuestra medida de privacidad desde distintos puntos de vista. En particular, la Sección IV-C investiga los fundamentos sobre los que se sustentan los métodos de maximización de la entropía, la minimización de la divergencia y la minimización de la ganancia de información. Para comprender estos argumentos, revisamos la propiedad de equipartición asintótica, el test de hipótesis y el lema de Stein.

IV-A. Criterio de Privacidad

Nuestro modelo matemático representa las *consultas* de usuario como v.a.'s que toman valores en un alfabeto común. Asumiremos que las consultas de usuario no son elaboradas o detalladas. En su lugar, éstas se referirán a un conjunto de categorías o temas, o de forma equivalente, podrán representar palabras clave en un conjunto indexable reducido. Por consiguiente, consideraremos que el alfabeto es finito. En concreto, asumiremos que las consultas toman valores en el alfabeto $\mathscr{X} = \{1, \ldots, n\}$ para algún $n \in \mathbb{Z}^+$.

Teniendo en cuenta estas consideraciones, definiremos p como la distribución de consultas de la *población*, q como la distribución real de un *usuario* en particular, y r como la distribución de las consultas *falsificadas* de ese usuario. Asimismo, consideraremos un parámetro de *redundancia* de consultas $0 \leqslant \rho \leqslant 1$, que será el ratio entre consultas falsificadas y consultas totales. De acuerdo con esto, definiremos la

distribución de consultas *aparente* del usuario s como la combinación convexa $(1-\rho)\,q+\rho\,r$, que será la distribución que en realidad observará el proveedor de servicios de información, o simplemente, un atacante de la privacidad. Un atacante será capaz de comprometer la privacidad de un usuario siempre que la distribución de consultas aparente de este usuario difiera de la distribución de consultas de la población.

Inspirados por los criterios de privacidad propuestos en [14]–[17], definimos el *riesgo de privacidad inicial* como la divergencia KL entre la distribución del usuario y la de la población, es decir, $\mathcal{R}_0 = \mathrm{D}(q \,\|\, p)$. De forma similar, definimos el *riesgo de privacidad final* $\mathcal{R}$ como la divergencia KL entre la distribución aparente y la distribución de la población, es decir,

$$\mathcal{R} = \mathrm{D}(s \,\|\, p) = \mathrm{D}((1-\rho)\,q + \rho\,r \,\|\, p).$$

IV-B. Compromiso Óptimo entre Privacidad de Consultas y Redundancia

Esta sección muestra una formulación del compromiso entre privacidad y redundancia para la generación de consultas falsas, que surge de la medida de privacidad presentada en la Sección IV-A. Partiendo de la definición de nuestro criterio de privacidad, supondremos que la población es suficientemente grande como para despreciar el impacto de la elección de r en p. De esta forma, definimos la función *privacidad-redundancia*

$$\mathcal{R}(\rho) = \min_{r} \ \mathrm{D}((1-\rho)\,q + \rho\,r \,\|\, p), \tag{1}$$

que representa el compromiso óptimo entre riesgo de privacidad de consultas y redundancia.

El término *mínimo* que aparece en la definición de la función privacidad-redundancia está justificado por el hecho de que el problema de optimización planteado implica una función acotada inferiormente y semi-continua inferiormente sobre un conjunto compacto, que es el símplex de probabilidad al que pertenece r.

Teniendo en cuenta esta formulación, conviene apreciar que es posible obtener resultados teóricos análogos para una definición alternativa del riesgo de privacidad, dada por la inversión de los argumentos de la divergencia KL. En la Sección IV-C2 se dan más detalles sobre esta formulación alternativa.

IV-C. Interpretación y Justificación

En esta sección interpretaremos y justificaremos la divergencia KL como criterio de privacidad en la definición de la función privacidad-redundancia. En concreto, examinaremos los argumentos en la literatura que abogan por la maximización de la entropía, y la minimización de la divergencia y la ganancia de información.

Antes de proceder a la interpretación y justificación de nuestro criterio de privacidad, querríamos comentar que, aunque nuestra propuesta surge de una cantidad de teoría de la información y resulta matemáticamente tratable, la adecuación de nuestra formulación está supeditada a la adaptación de los criterios optimizados, que a su vez depende de varios factores tales como la propia aplicación, el modelo de adversario y los mecanismos en contra de la privacidad que se hayan contemplado. Las interpretaciones y justificaciones que aquí se detallan tienen por objeto ayudar a los diseñadores y usuarios de sistemas a evaluar la adecuación de nuestra propuesta a una aplicación específica de recuperación de información.

Asimismo, querríamos poner énfasis en que, a pesar de que nuestro criterio de privacidad se basa en una cantidad fundamental de teoría de la información, la convergencia de estos dos campos en absoluto es nueva. De hecho, el trabajo de Shannon en los años cincuenta ya introdujo el concepto de *equivocación* como la entropía condicional de un mensaje privado dada la observación de un criptograma [20], utilizada más tarde en la formulación del problema *wiretap channel* [21], [22] como una medida de confidencialidad. Del mismo modo, podemos mencionar la interpretación basada en teoría de la información de la divergencia entre las distribuciones a priori y a posterior, denominada *ganancia de información promedio* en algunos campos de estadística [23], [24]. Asimismo, estudios recientes [17] reafirman la adecuación y aplicabilidad del concepto de entropía como medida de privacidad, tal y como comentamos en la Sección II.

IV-C1. Maximización de la Entropía: Nuestra primera interpretación está basada, de hecho, en la idea de que la entropía de Shannon se puede considerar como un caso particular del criterio propuesto en este artículo. Para comprender esta conexión, suponga que la distribución de consultas de la población es la distribución uniforme u sobre el alfabeto $\mathscr{X}$, es decir, que $u_i = 1/n$ para todo $i \in \mathscr{X}$. En este supuesto, el riesgo de privacidad se puede expresar como

$$\mathrm{D}((1-\rho)\,q + \rho\,r \,\|\, u) = \ln n - \mathrm{H}((1-\rho)\,q + \rho\,r).$$

Por tanto, minimizar la divergencia KL es equivalente a maximizar la entropía de la distribución de consultas aparente del usuario:

$$\mathcal{R}(\rho) = \ln n - \max_{r} \ \mathrm{H}((1-\rho)\,q + \rho\,r).$$

Esta equivalencia nos conduce a las siguientes dos implicaciones. En primer lugar, el criterio de privacidad $\mathrm{H}((1-\rho)\,q+\rho\,r)$ es una medida de *ganancia* de privacidad, más que de *riesgo* de privacidad. En segundo lugar, se trata de una medida de privacidad *absoluta*, en contraste con nuestro criterio más general, en el sentido que es una métrica *relativa* a cualquier distribución de referencia.

El hecho de considerar esta medida absoluta de ganancia de privacidad permite acercarnos a los fundamentos sobre los que se apoyan los métodos de *maximización de la entropía*. Algunos de estos argumentos están relacionados con el mayor número de permutaciones con repetición asociado a una distribución empírica [25]. Sin embargo, el argumento más apropiado para la justificación de la maximización de la entropía, considerada como una medida de privacidad,

viene dado por la propiedad de equipartición asintótica (AEP, *asymptotic equipartition property*) [19, §3].

Suponga una secuencia $X_1, \ldots, X_k$ de k consultas i.i.d.'s, que toman valores en $\mathscr{X}$, y son generadas de acuerdo con la distribución de consultas aparente del usuario $s = (1 - \rho)\, q + \rho\, r$. Para k suficientemente grande, la AEP sostiene que es muy probable que la secuencia de consultas $x_1, \ldots, x_k$ pertenezca a un subconjunto $\mathscr{T}^{(k)}$ del conjunto de todas las posibles secuencias, denominado *conjunto típico*, que satisface estas propiedades: la probabilidad de este conjunto es aproximadamente 1, todos los elementos son casi equiprobables, y el número de elementos es prácticamente $e^{k\,\mathrm{H}(s)}$. Resulta que la entropía está acotada superiormente por $\ln n$, como consecuencia de la no negatividad de la divergencia KL, y alcanza su valor máximo cuando la distribución aparente es la distribución uniforme. A partir de esta observación, podemos deducir que la distribución uniforme maximiza el conjunto típico $\mathscr{T}^{(k)}$ y, cuando esto sucede, éste se convierte en el conjunto de todos los posibles resultados, conteniendo n^k secuencias. Puesto que $\mathrm{H}(s)$ caracteriza completamente esta aproximación, cualquier medida de privacidad con sentido acabaría siendo básicamente equivalente a ésta.

Teniendo en cuenta esta conexión entre entropía y tamaño del conjunto típico, ahora describiremos la siguiente amenaza de privacidad. Suponga que un atacante intenta adivinar una secuencia de k consultas de un usuario en particular a partir de la observación de secuencias previas. Cuanto mayor sea la entropía $\mathrm{H}(s)$ de la distribución de consultas aparente del usuario, mayor será el tamaño del conjunto típico $\mathscr{T}^{(k)}$ de secuencias posibles e igualmente probables de k consultas, y mayor la probabilidad de que la secuencia a adivinar sea significativamente diferente de las anteriores. Este escenario nos permite concluir que los métodos de maximización de la entropía contribuyen ampliamente a la protección de la privacidad del usuario.

IV-C2. Minimización de la Divergencia: En la sección anterior examinamos los argumentos que abogan por la maximización de la entropía. En esta sección, recurriremos al lema de Stein, revisado en la Sección III, para nuestra interpretación de la divergencia como falsos positivos y falsos negativos. En concreto, describiremos un escenario en el que un atacante utiliza test de hipótesis para comprometer la privacidad del usuario.

En el resto de la sección, consideraremos nuestro criterio de privacidad en su sentido más amplio, es decir, la distribución de consultas del usuario no se comparará necesariamente, en términos de la divergencia KL, con la distribución uniforme.

Nuestra interpretación contempla el escenario en el que un atacante conoce, o es capaz de estimar, la distribución de consultas aparente s de un usuario determinado. Además, suponemos que el atacante observa una secuencia de k consultas i.i.d.'s, e intenta adivinar si éstas han sido generadas por ese usuario o no. Exactamente, el atacante considera el test de hipótesis binario entre dos alternativas: si las consultas se han producido de acuerdo con la distribución aparente del usuario s, hipótesis $\mathscr{U}$, o la distribución general de la población p, hipótesis $\mathscr{P}$.

Llegados a este punto, un atacante podría llevar a cabo dos estrategias mutuamente excluyentes. La primera estrategia considera que el atacante está interesado en acotar la probabilidad de un falso negativo $\mathrm{P}(\mathscr{P}|\mathscr{U})$, dado que su objetivo es que el usuario no pase desapercibido. A partir del lema de Stein, encontramos que la probabilidad $\mathrm{P}(\mathscr{P}|\mathscr{U})$ de un falso positivo es aproximadamente $e^{-k\,\mathrm{D}(s\,\|\,p)}$ para k grande. Por consiguiente, la minimización de $\mathrm{D}(s\,\|\,p)$ en la definición de la función privacidad-redundancia (1) implica la maximización del exponente en la tasa de error de falsos positivos. Dicho de otra forma, la distribución óptima de consultas falsas r^* frustra a un atacante en su esfuerzo por reconocer a un usuario de entre la población, y por tanto, comprometer la privacidad del usuario.

Más que fijar la probabilidad de un falso negativo, ahora el objetivo del atacante es minimizar la probabilidad de error global

$$P_{\mathrm{T}} = \mathrm{P}(\mathscr{U})\,\mathrm{P}(\mathscr{P}|\mathscr{U}) + \mathrm{P}(\mathscr{P})\,\mathrm{P}(\mathscr{U}|\mathscr{P}).$$

Aprovechándose del hecho de que la actividad de la población global es mucho mayor que la de un único usuario, el atacante está interesado en acotar $\mathrm{P}(\mathscr{U}\,|\mathscr{P})$, y hacer lo posible para minimizar $\mathrm{P}(\mathscr{P}|\mathscr{U})$. Resulta que la probabilidad de un falso negativo dado por el lema de Stein es aproximadamente $e^{-\,\mathrm{D}(p\,\|\,s)}$, lo que justifica una definición alternativa de la función privacidad-redundancia dada por la inversión de los dos argumentos de la divergencia KL. De acuerdo con esta observación, la estrategia de falsificación de consultas r^* que minimiza $\mathrm{D}(p\,\|\,s)$, conduce a la maximización de la probabilidad de error global del atacante y contribuye a proteger la privacidad del usuario.

A modo de aclaración, nos gustaría destacar que, a pesar de que esta definición alternativa resulta oportuna en el último escenario propuesto, nosotros creemos que la formulación original es más apropiada, ya que incluye, como caso particular, los métodos de maximización de la entropía descritos en la Sección IV-C1.

IV-C3. Minimización de la Ganancia de Información: Una vez analizados los principales argumentos en pro de la maximización de la entropía y la minimización de la divergencia, ahora estableceremos una conexión entre nuestro criterio de privacidad y el criterio propuesto en [16].

Considere $p_{Q|U}(q|u)$ la distribución de consultas del usuario u, donde U es una variable aleatoria que identifica a un usuario en particular y toma el valor u. Asimismo, Q es una variable aleatoria que representa una consulta en particular, y toma el valor q.

Sea $p_Q(q)$ la distribución de probabilidad sin condicionar que modela la distribución de consultas de la población. Naturalmente, $p_U(u)$ sería la probabilidad de usuario, posiblemente ponderada por su actividad. En esta notación, nuestra medida de riesgo de privacidad para el usuario u se puede escribir como $\mathrm{D}(p_{Q|U}(\cdot|u)\|p_Q)$. De forma similar, podemos aplicarla

para redefinir el concepto de t-closeness. Una distribución satisface t-closeness si y sólo si $\mathrm{D}(p_{Q|U}(\cdot|u)\|p_Q) \leq t$ para todos los valores u de U, lo que sugiere medir el riesgo de privacidad como un máximo sobre divergencias. Inspirados por t-closeness, [16] presenta un planteamiento más interesante en el sentido que nos permite conectar con la ganancia de información promedio. En concreto, el criterio de privacidad propuesto en [16] es la divergencia KL condicional

$$\mathrm{D}(p_{Q|U}\|p_Q) = \mathrm{E}_U\,\mathrm{D}(p_{Q|U}(\cdot|U)\|p_Q),$$

es decir, la información mutua entre Q y U, o de forma equivalente, el promedio entre usuarios del criterio de privacidad definido en este artículo. En contraste con este criterio, nuestra medida de privacidad contempla un único usuario, pero podría, en principio, generalizarse a escenarios multiusuarios en una futura propuesta.

V. Conclusiones

Existe una gran variedad de propuestas para PIR, considerado aquí en el sentido más amplio del término. Dentro de estas soluciones, la generación de consultas falsas surge como una estrategia simple en términos de requisitos de infraestructura, ya que los usuarios no necesitan una entidad externa en la que confiar. Sin embargo, esta solución plantea un compromiso entre la privacidad y el coste de la sobrecarga de tráfico y procesado.

Nuestra principal contribución es un criterio de privacidad basado en teoría de la información para la falsificación de consultas en PIR, que emerge de la formulación del compromiso entre privacidad y redundancia. Inspirados por el trabajo en [16], medimos el riesgo de privacidad como la divergencia KL entre la distribución de consultas aparente del usuario, que contiene consultas falsas, y la de la población. Nuestra formulación contempla, como caso especial, la maximización de la entropía de la distribución del usuario.

En este artículo justificamos nuestro criterio de privacidad al interpretarlo desde distintas perspectivas, y al conectarlo con los argumentos en la literatura que abogan por la maximización de la entropía, la minimización de la divergencia y la minimización de la ganancia de información. Nuestras interpretaciones están basadas en la AEP, el test de hipótesis y el lema de Stein, y el criterio de ganancia de información promedio propuesto en [16].

Aunque nuestra propuesta surge de una medida de teoría de la información y resulta matemáticamente tratable, la adecuación de nuestra formulación está supeditada a la adaptación de los criterios optimizados, que a su vez depende de varios factores tales como la propia aplicación, la estadística de consultas de los usuarios, la sobrecarga de red y de procesado provocados por la consultas falsas, el modelo de adversario y los mecanismos en contra de la privacidad que se hayan contemplado.

Agradecimientos

Este trabajo ha sido financiado en parte por el gobierno español mediante los proyectos CONSOLIDER INGENIO 2010 CSD2007-00004 “ARES” y TSI2007-65393-C02-02 “ITAC”, y por el gobierno catalán bajo la subvención 2009 SGR 1362.

Referencias

[1] D. Fallows, “Search engine users,” Pew Internet and American Life Project, Tech. Rep., Jan. 2005.

[2] S. Fox and S. Jones, “The social life of health information,” Pew Internet and American Life Project, Tech. Rep., Jun. 2009.

[3] C. C. M. F. Mokbel and W. G. Aref, “The new casper: query processing for location services without compromising privacy,” in *Proc. Int. Conf. on Very Large Data Bases*. VLDB J., 2006, pp. 763–774.

[4] M. Duckham, K. Mason, J. Stell, and M. Worboys, “A formal approach to imperfection in geographic information,” *Comput., Environ., Urban Syst.*, vol. 25, no. 1, pp. 89–103, 2001.

[5] R. Ostrovsky and W. E. Skeith III, “A survey of single-database PIR: Techniques and applications,” in *Proc. Int. Conf. Practice, Theory Public-Key Cryptogr. (PKC)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 4450. Beijing, China: Springer-Verlag, Sep. 2007, pp. 393–411.

[6] B. S. Y. Elovici and A. Maschiach, “A new privacy model for hiding group interests while accessing the web,” in *Proc. ACM Workshop on Privacy in the Electron. Society*. ACM, 2002, pp. 63–70.

[7] B. Shapira, Y. Elovici, A. Meshiach, and T. Kuflik, “PRAW – The model for PRivAte Web,” *J. Amer. Soc. Inform. Sci., Technol.*, vol. 56, no. 2, pp. 159–172, 2005.

[8] C. Soghoian, “The problem of anonymous vanity searches,” *I/S: J. Law, Policy Inform. Soc. (ISJLP)*, Jan. 2007.

[9] P. Samarati and L. Sweeney, “Protecting privacy when disclosing information: k-Anonymity and its enforcement through generalization and suppression,” SRI Int., Tech. Rep., 1998.

[10] X. Sun, H. Wang, J. Li, and T. M. Truta, “Enhanced p-sensitive k-anonymity models for privacy preserving data publishing,” *Trans. Data Privacy*, vol. 1, no. 2, pp. 53–66, 2008.

[11] A. Machanavajjhala, J. Gehrke, D. Kiefer, and M. Venkitasubramanian, “l-Diversity: Privacy beyond k-anonymity,” in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, Atlanta, GA, Apr. 2006, p. 24.

[12] H. Jian-min, C. Ting-ting, and Y. Hui-qun, “An improved V-MDAV algorithm for l-diversity,” in *Proc. IEEE Int. Symp. Inform. Processing (ISIP)*, Moscow, Russia, May 2008, pp. 733–739.

[13] J. Domingo-Ferrer and V. Torra, “A critique of k-anonymity and some of its enhancements,” in *Proc. Workshop Privacy, Security, Artif. Intell. (PSAI)*, Barcelona, Spain, 2008, pp. 990–993.

[14] N. Li, T. Li, and S. Venkatasubramanian, “t-Closeness: Privacy beyond k-anonymity and l-diversity,” in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, Istanbul, Turkey, Apr. 2007, pp. 106–115.

[15] D. Rebollo-Monedero, J. Forné, and J. Domingo-Ferrer, “From t-closeness to PRAM and noise addition via information theory,” in *Privacy Stat. Databases (PSD)*, ser. Lecture Notes Comput. Sci. (LNCS). Istambul, Turkey: Springer-Verlag, Sep. 2008, pp. 100–112.

[16] ——, “From t-closeness-like privacy to postrandomization via information theory,” *IEEE Trans. Knowl. Data Eng.*, Oct. 2009. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TKDE.2009.190

[17] C. Díaz, S. Seys, J. Claessens, and B. Preneel, “Towards measuring anonymity,” in *Proc. Workshop Privacy Enhanc. Technol. (PET)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 2482. Springer-Verlag, Apr. 2002.

[18] C. Díaz, “Anonymity and privacy in electronic services,” Ph.D. dissertation, Katholieke Univ. Leuven, Dec. 2005.

[19] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York: Wiley, 2006.

[20] C. E. Shannon, “Communication theory of secrecy systems,” Bell Syst., Tech. J., 1949.

[21] A. Wyner, “The wiretap channel,” Bell Syst., Tech. J. 54, 1975.

[22] I. Csiszár and J. Körner, “Broadcast channels with confidential messages,” *IEEE Trans. Inform. Theory*, vol. 24, pp. 339–348, May 1978.

[23] P. M. Woodward, “Theory of radar information,” in *Proc. London Symp. Inform. Theory, Ministry of Supply*, London, UK, 1950, pp. 108–113.

[24] D. V. Lindley, “On a measure of the information provided by an experiment,” *Annals Math. Stat.*, vol. 27, no. 4, pp. 986–1005, 1956.

[25] E. T. Jaynes, “On the rationale of maximum-entropy methods,” *Proc. IEEE*, vol. 70, no. 9, pp. 939–952, Sep. 1982.

Microagregación para el k-anonimato en registros de buscadores Web

Guillermo Navarro-Arribas*, Vicenç Torra*, Arnau Erola†, Jordi Castellà-Roca†
* IIIA, Institut d'Investigació en Intel.·ligència Artificial -
CSIC, Consejo Superior de Investigaciones Científicas,
Email: {guille, vtorra}@iiia.csic.es
† Departament d'Enginyeria Informàtica i Matemàtiques,
UNESCO Chair in Data Privacy, Universitat Rovira i Virgili,
Email: {arnau.erola, jordi.castella}@urv.cat

***Resumen*—La conversión de los registros de búsqueda en anónimos es un proceso importante y necesario previo a la publicación de dichos datos. Ésto asegura la privacidad de los usuarios, un problema que ya se ha dado en los registros publicados por importantes compañías. El artículo presenta la conversión de los registros de un buscador Web en anónimos utilizando microagregación. Nuestra propuesta garantiza el k-anonimato a nivel de usuario preservando la utilidad de los datos.**

I. Introducción

Los registros (o *logs*) generados por un buscador Web (WSE) son una fuente interesante de información para investigadores o compañías de marketing, pero al mismo tiempo, su publicación supone una amenaza contra la privacidad de los usuarios de los WSEs. Existe un caso conocido en el cual la publicación de este tipo de información, con un nivel de anonimato pobre, ha permitido la reidentificación de algunos usuarios. AOL publicó unos registros de su buscador, supuestamente anónimos, con el propósito de ayudar la comunidad científica y acabó con un daño irreparable hacia la privacidad de sus usuarios y daños contra la propia empresa con diversas quejas y denuncias [8], [17].

La dificultad de la convertir en anónimos los registros de búsqueda es conseguir un buen balance entre privacidad y utilidad. Existen algunas propuestas para conseguir anonimato este tipo de datos [3], pero generalmente se limitan a borrar información específica de los registros. Es más, técnicas de control de revelación estadística (*statistical disclosure control*, SDC) no han sido aplicadas a este tipo de datos hasta muy recientemente [18], [10].

En este artículo proponemos el uso de la microagregación para proporcionar anonimato en registros de búsqueda. Nuestra propuesta consigue un alto grado de privacidad, proporcionando k-anonimato a nivel de usuario, y preservando, hasta cierto punto, la utilidad de los datos.

El artículo está organizado como sigue. La sección II introduce el problema y nuestras motivaciones. En la sección III presentamos nuestra propuesta de microagregación. La sección IV contiene una evaluación de la propuesta en términos de privacidad y utilidad de datos, y finalmente la sección V presenta las conclusiones.

II. Motivaciones

Un registro de búsqueda de un WSE está compuesto por líneas con la estructura:

$$(id, q, t, r, u)$$

donde id es el identificador de usuario, q son los términos de búsqueda, t es el timestamp, u es la URL seleccionada por el usuario después de la búsqueda, y r es la posición de la URL seleccionada en el resultado de la búsqueda. Este formato corresponde a los registros publicados por AOL en el 2006. La Figura 1 muestra algunas de estas líneas reales de los datos de AOL. La información proporcionada en estos registros es la misma que la de los registros publicados por AllTheWeb [11], y muy parecida a otros registros publicados por Excite [13] o AltaVista [12]. Normalmente se considera como un formato de registro de búsqueda genérico.

Nuestro trabajo se centra en los registros de AOL. Hemos de tener en cuenta que estos datos han sido convertidos a anónimos con técnicas bastante pobres. La URL seleccionada (u) es cortada a nivel de nombre de dominio y la información sensible de los términos de búsqueda ha sido eliminada (números de la seguridad social, ...) [25]. El identificador de usuario (id) es un identificador único, su anonimato se consigue mediante una función hash o técnicas similares. Se ha demostrado que a pesar de esta protección los usuarios pueden ser identificados [2]. Es más, el uso de técnicas de hash aplicadas a los términos de búsqueda puede ser vulnerable al análisis de frecuencia [15].

Se han desarrollado otras técnicas para conseguir anonimato en registros de búsqueda como la eliminación de búsquedas poco frecuentes [1], o propuestas más sofisticadas para eliminar búsquedas concretas que permitan preservar un nivel de privacidad aceptable [20], o escoger las búsquedas que se pueden publicar [14].

El trabajo presentado en este artículo parte de una propuesta inicial [18], que has sido mejorada tanto en términos de eficiencia como en los resultados obtenidos.

II-A. k-anonimato a nivel de usuario

Un principio muy utilizado en el control de revelación de datos estadísticos (SDC) es el k-anonimato [21], [22],

```
24963762 myspace codes  2006-05-31 23:00:52  2  http://www.myspace-codes.com
24964082 bank of america  2006-05-31 19:41:07  1  http://www.bankofamerica.com
24967641 donut pillow  2006-05-31 14:08:53
24967641 dicontinued dishes  2006-05-31 14:29:38
24969374 orioles tickets  2006-05-31 12:31:57  2  http://www.greatseats.com
24969374 baltimore marinas  2006-05-31 12:43:40
```

Figura 1: Ejemplo de registro de un buscador Web.

que especifica que cada consulta de los datos anónimos debe retornar como mínimo k registros iguales. Conseguir k-anonimato a nivel de usuario en los registros de búsqueda es el principal objetivo de nuestra propuesta, ya que previene la reidentificación de los usuarios que era posible en los registros inicialmente publicados por AOL [2].

Para conseguir k-anonimato a nivel de usuario aplicamos microagregación a los registros, lo que permite no tener que borrar explícitamente ninguno. Para poder microagregar los registros de búsqueda a nivel de usuario, consideramos como un solo registro todos los registros de un mismo usuario. Es decir, hay un registro por usuario, y cada registro contiene todos los registros de dicho usuario, que son tratados como un todo en el proceso de protección.

Un importante inconveniente de nuestra propuesta es que se pierde algo de información en los datos protegidos. Como en casos similares siempre existe un balance entre privacidad y utilidad. En nuestro caso, mostramos que, aún consiguiendo un nivel de privacidad muy alto, los datos siguen preservando suficiente utilidad para ser usados en procesos de data-mining o análisis estadístico.

III. Microagregación de registros de búsqueda

Para poder aplicar microagregación a los registros de búsqueda debemos definir el proceso de microagregación. En las secciones siguientes introducimos la microagregación y como la utilizamos para proteger registros de búsqueda.

III-A. Microagregación

La microagregación es una técnica de control de revelación estadística (SDC), que proporciona privacidad mediante la agrupación de los datos en pequeños clústers y sustituyendo luego los datos originales por los representantes (centroides) del clúster correspondiente.

La privacidad se consigue porqué cada clúster tiene un número mínimo predefinido de elementos: hay al menos k registros con el mismo valor. Todos los registros del clúster cambian su valor por el del centroide del clúster. La constante k es un parámetro que controla el nivel de privacidad del método. A medida que aumentamos k conseguimos más privacidad en los datos protegidos.

La microagregación fue originalmente propuesta para atributos numéricos [4], aunque más tarde fue ampliada a otros dominios. Por ejemplo, a datos categóricos en [23] (ver también [7]), o a entornos con restricciones en [24].

Desde el punto de vista operacional, la microagregación se define en términos de partición y agregación:

- **Partición.** Los registros se dividen en clústers, de manera que cada clúster tenga al menos k registros.
- **Agregación.** Para cada clúster, se calcula un representante (centroide), y se sustituyen los registros originales por dicho representante.

Desde un punto de vista formal, la microagregación se puede definir como un problema de optimización con algunas restricciones. Damos una formalización a continuación utilizando u_{ij} para denotar la partición de registros en el conjunto de datos original X. Esto es, $u_{ij} = 1$ si el registro j está asignado al clúster i. Sea v_i el representante del clúster i; la formulación general de la microagregación con g clústers y dado el parámetro k es:

$$\begin{aligned} &\text{Minimizar} \quad SSE = \textstyle\sum_{i=1}^{g}\sum_{j=1}^{n} u_{ij}(d(x_j, v_i))^2 \\ &\text{Sujeto a} \quad \textstyle\sum_{i=1}^{g} u_{ij} = 1 \text{ para todo } j = 1, \ldots, n \\ &\qquad\qquad 2k \geq \textstyle\sum_{j=1}^{n} u_{ij} \geq k \text{ para todo } i = 1, \ldots, g \\ &\qquad\qquad u_{ij} \in \{0, 1\} \end{aligned}$$

Para datos numéricos, generalmente $d(x, v)$ corresponde a la distancia Euclídea. En el caso general, cuando se consideran los atributos $\mathbf{V} = (V_1, \ldots, V_s)$, x y v son vectores y d pasa a ser $d^2(x, v) = \sum_{V_i \in \mathbf{V}} (x_i - v_i)^2$. Además, suele ser común en datos numéricos definir v_i como la media aritmética de los registros del clúster. Es decir, $v_i = \sum_{j=1}^{n} u_{ij} x_i / \sum_{j=1}^{n} u_{ij}$. Dado que la solución a este problema cuando se considera más de una variable a la vez (microagregación multivariable) es NP-Hard [19] se han desarrollado métodos heurísticos.

MDAV [5] (*Maximum Distance to Average Vector*) es uno de estos algoritmos heurísticos. Se encuentra descrito con detalle en el Algoritmo 1, donde se aplica a un conjunto de datos X con n registros y A atributos. La implementación de MDAV para datos categóricos se da en [7].

Nótese que cuando todas las variables se consideran al mismo tiempo, la microagregación es una manera de implementar k-anonimato [21], [22].

III-B. Distancia y agregación de registros de búsqueda

Para microagregar registros de búsqueda a nivel de usuario, necesitamos definir una distancia adecuada para particionar los datos y un operador de agregación para el cálculo del centroide.

Denotamos cada usuario id_i como:

$$q(id_i) = (id_i, \varphi^i)$$

donde $\varphi^i = (\varphi^i_1, \varphi^i_2, \varphi^i_3, \ldots)$ es el vector de búsquedas hechas por el usuario id_i. Es decir, φ^i_j corresponde a la búsqueda j del usuario id_i, y esta compuesta por $\varphi^i_j = \{t^i_j, r^i_j, u^i_j, \phi^i_j\}$, donde $\phi^i_j = (\mu_0, \mu_1, \mu_2, \ldots)$ es la cadena de búsqueda (términos

Algorithm 1: Algoritmo MDAV

Data: X: conjunto de datos original, k: entero
Result: X': conjunto de datos protegido

begin
  while $(|X| \geq 3 * k)$ **do**
    Calcular el registro medio $\bar{x}$ de todos los registros en X;
    Seleccionar el registro x_r más distante al registro medio $\bar{x}$;
    Formar un clúster entorno a x_r. El clúster contiene x_r junto a los $k-1$ registros más cercanos a x_r;
    Borrar estos registros del conjunto X;
    Seleccionar el registro x_s más distante del registro x_r;
    Formar un un clúster entorno a x_s. El clúster contiene x_s junto a los $k-1$ registros más cercanos a x_s;
    Borrar estos registros del conjunto X;
  if $(|X| >= 2 * k)$ **then**
    Calcular el registro medio $\bar{x}$ de todos los registros en X;
    Seleccionar el registro x_r más distante al registro medio $\bar{x}$;
    Formar un clúster entorno a x_r. El clúster contiene x_r junto a los $k-1$ registros más cercanos a x_r;
    Borrar estos registros del conjunto X;
  Formar un clúster con los registros restantes;
end

de búsqueda introducidos por el usuario). También utilizamos $|\varphi^i|$ para denotar el número de búsquedas del usuario id_i, y $|\phi^i_j|$ para el número de términos de búsqueda (palabras) en la búsqueda j del usuario id_i.

Un paso previo a la microagregación es la normalización de los datos numéricos: timestamp, posición, número de búsquedas por usuario, y número de términos por búsqueda. Dicha normalización se hace en el intervalo $[0, 1]$. El número de búsquedas normalizado del usuario id_i se denota como $\overline{|\varphi^i|}$, y el número de términos normalizado en la búsqueda φ^i_j como $\overline{|\phi^i_j|}$.

III-B1. Distancia: La distancia se calcula como la agregación de varias funciones de distancia para cada par de usuarios. Definimos la siguientes funciones de distancia:

- $d_{euclid}(x, y) = \sqrt{(x-y)^2}$: distancia euclídea que se utiliza para la posición r.
- $d_t(t_i, t_j)$: distancia entre dos timestamps t_1, t_2, como la distancia euclídea de su representación en *UNIX epoch*.
- $d_u(u_i, u_j)$: distancia entre dos nombres de dominio (URL clicada). Dados dos nombres de dominios: $X = x_n.....x_0$, y $Y = y_m.....y_0$, y asumiendo que $m \geq n$, la distancia es:

$$d_u(X, Y) = \sum_{i=0}^{m} w_i \alpha_i$$

donde $w_i = 2^{m-i}/(2^m - 1)$ y $\alpha_i = 0$ si $x_i = y_i$ (case-insensitive string equality) o 1 en caso contrario. Es decir, d_u es una media con pesos de α_i, donde consideramos más importante la parte derecha del nombre de dominio.
- $d_{lev}(x, y)$: la distancia de Levenshtein (o *edit distance*) normalizada entre dos cadenas de caracteres x, y.
- $d_\phi(\phi_i, \phi_j)$: distancia entre dos cadenas de búsqueda (términos introducidos por el usuario) como:

$$d_\phi(\phi_i, \phi_j) = \frac{1}{2}\left(\frac{\overline{|\phi_i|} + \overline{|\phi_j|}}{2} + d_{\mathcal{H}}(\phi_i, \phi_j)\right)$$

donde $d_{\mathcal{H}}$ es la distancia de Hausdorff definida en el espacio métrico (μ, d_{lev}), donde μ es el conjunto de todas las palabras μ_i. Cada búsqueda se representa como un conjunto de palabras o términos $\phi_i = \{\mu^i_1, \mu^i_2, \ldots\}$ y se utiliza la distancia Levenshtein para compararlos. De esta manera tenemos que:

$$d_{\mathcal{H}}(\phi_1, \phi_2) = \text{máx}(I_{\mathcal{H}}(\phi_1, \phi_2), I_{\mathcal{H}}(\phi_2, \phi_1))$$

donde

$$I_{\mathcal{H}}(\phi_1, \phi_2) = \max_{\mu_i \in \phi_1} \min_{\mu_j \in \phi_2} d_{lev}(\mu_i, \mu_j)$$

En este caso d_ϕ tiene en cuenta la similitud de los términos entre cadenas de búsqueda, pero también considera el tamaño de la cadena de búsqueda, algo que la distancia de Hausdorff no considera por sí sola.
- $d_\varphi(\varphi_i, \varphi_j)$: distancia entre dos búsquedas de la forma $\varphi_i = (t_i, r_i, u_i, \phi_i)$, como la media de las distancias correspondientes:

$$d_\varphi(\varphi_1, \varphi_2) = \frac{1}{4}(d_t(t_1, t_2), d_{euclid}(r_1, r_2), d_u(u_1, u_2), d_\phi(\phi_1, \phi_2))$$

Dadas las funciones de distancia anteriores, la distancia final entre dos usuarios se calcula como:

$$d(q(id_1), q(id_2)) = \frac{1}{3}\left(\overline{|\varphi^1|} + \overline{|\varphi^2|} + d_{\mathcal{H}}(\varphi^1, \varphi^2)\right)$$

donde $d_{\mathcal{H}}$ es la distancia de Hausdorff en el espacio métrico (φ, d_φ), y donde φ es el conjunto de todas las búsquedas φ^i.

III-B2. Agregación de usuarios: Para determinar el centroide de un un clúster de registros de usuarios, calculamos su agregación ($\mathbb{C}$) como la agregación de cada parte de las búsquedas del usuario:

$$\mathbb{C}(q(id_1), \ldots, q(id_k)) = (id', \mathbb{C}_\varphi(\varphi^1, \ldots, \varphi^k))$$

donde, id' es un identificador temporal para el centroide que será sustituido por el identificador original del usuario en los datos protegidos.

De la misma manera, la agregación de búsquedas $\mathbb{C}_\varphi$ se define como:

$$\begin{aligned}\mathbb{C}_\varphi(\varphi^1,\dots\varphi^k) &= \mathbb{C}_\varphi\left((\varphi_1^1,\dots,\varphi_{|\varphi^1|}^1),\dots,(\varphi_1^k,\dots,\varphi_{|\varphi^k|}^k)\right)\\ &= \varphi^*\\ &= (\varphi_1^*,\dots,\varphi_{|\varphi^*|}^*)\end{aligned}$$

El centroide φ^* está compuesto de las cadenas de búsqueda del clúster φ^i para $i = 1\dots k$. Por cada vector de búsqueda original φ^i, eso es, todas las búsquedas del usuario i, tomamos un sub-vector $\varphi^{*,i}$ de búsquedas tal que:

$$|\varphi^{*,i}| = \frac{|\varphi^*|\cdot|\varphi^i|}{\sum_{j=1}^k |\varphi^j|}$$

Estas búsquedas, $\varphi^{*,i} = (\varphi_1^{*,i},\dots,\varphi_{|\varphi^{*,i}|}^{*,i})$, preservan la frecuencia de cadenas de búsqueda del original φ^i. De manera más formal, dada una función de frecuencia f sobre cadenas de búsqueda, se tiene que cumplir que,

$$f(\varphi_q^{*,i}) \simeq f(\varphi^i)$$

donde,

$$f(\varphi_q^{*,i}) = \frac{|\{\varphi \mid \varphi \doteq \varphi_q^{*,i},\ \varphi \in \varphi^{*,i}\}|}{|\varphi^{*,i}|}$$

donde $\varphi_i \doteq \varphi_j$ si las cadenas de búsqueda de ambas peticiones son iguales, es decir, si y solo si $\phi_i = \phi_j$.

Las otras partes de las peticiones se agregan utilizando la media aritmética para el rango, y el timestamp, y generalizando la URL a la parte más común (sub-dominio).

III-C. Ejemplo de agregación de usuarios

A continuación mostramos un ejemplo muy sencillo y simplificado de la agregación de búsquedas de usuario anteriormente propuesta. Consideramos tres usuarios u_1, u_2, y u_3. Para simplificar el ejemplo únicamente consideramos las cadenas de búsqueda de cada petición.

u_1	all blues
u_1	blues for alice
u_1	the blues and beyond
u_2	all blues
u_2	alice
u_3	on green dolphin street
u_3	miles davis blues

$\Rightarrow$

$u^* = \mathbb{C}(u_1,u_2,u_3)$	
u^*	all blues
u^*	on green dolphin street
u^*	the blues and beyond

Cuadro I: Ejemplo de agregación de búsquedas de usuario.

El Cuadro I muestra las búsquedas de los tres usuarios, así como el resultado de su agregación.

IV. Evaluación

Para evaluar nuestra propuesta hemos probado la microagregación con datos reales de los registros de AOL hechos públicos en 2006, que corresponden a las consultas realizadas por 650,000 usuarios durante tres meses. Para nuestro estudio, hemos seleccionado aleatoriamente 1,000 usuarios de los registros, que corresponden a 55,666 líneas de consultas de los registros.

En las siguientes secciones, medimos la privacidad conseguida con nuestro método y la utilidad de los datos protegidos.

IV-A. Porcentaje de Perfil Expuesto

Para cada usuario id disponemos de su conjunto original de consultas φ y sus correspondientes protegidas mediante nuestro sistema de microagregación φ'. Con la finalidad de verificar que nuestro método protege las consultas de los usuarios ofreciendo k-anonimato hemos usado el *Porcentaje de Perfil Expuesto* (PPE) [9] que se define de la forma siguiente:

$$PPE \quad = \quad \frac{I(\varphi,\varphi')}{H(\varphi)}\cdot 100$$

donde $H(\varphi)$ es la entropía del conjunto original de consultas, y $I(\varphi,\varphi')$ es la información mutua entre φ y φ'. Nótese que φ y φ' pueden ser observadas como dos variables discretas.

El PPE mide el porcentaje de información del usuario que se ve expuesto cuando φ' es revelada. Así, la información del usuario se calcula como la entropía de φ, y la información mutua proporciona una medida de la información que φ' proporciona sobre φ, es decir, cuando φ' es conocido, en cuanto se reduce la incertidumbre sobre φ.

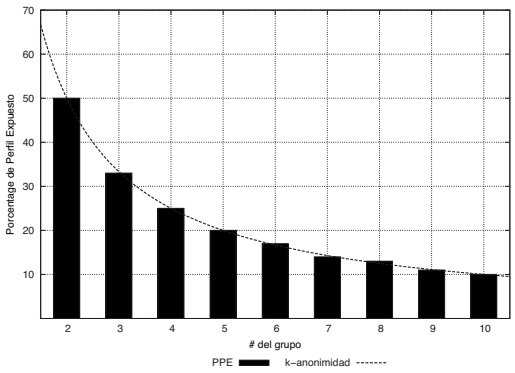


Figura 2: PPE medio para $k \in \{2,\dots,10\}$.

Hemos microagregado los 1,000 usuarios de los registros de AOL para $k \in \{2,\dots,50\}$. A continuación hemos calculado el PPE para cada usuario y valor de k.

La Figura 2 muestra para $k = \{2,\dots,10\}$ el nivel de privacidad teórico, es decir, el nivel de k-anonimato, y la media del PPE obtenido. Así, podemos ver que nuestro método ofrece k-anonimato, ya que el PPE teórico y el obtenido son muy parecidos.

IV-B. Pérdida de información

La primera medida de utilidad para datos categóricos fue presentada en [6]. Los autores proponen una medida basada en la entropía de los datos para evaluar la pérdida de información en SDC. En la misma línea, en [16] se propone medir la pérdida de información como:

$$ILR = \frac{|entropia\_original - entropia\_nueva|}{entropia\_original}\cdot 100$$

Hemos escogido el ILR (*Information Loss Ratio*) para evaluar la utilidad de nuestra propuesta con los mismos ficheros obtenidos previamente cuando calculamos el PPE. Así, tenemos $1{,}000$ usuarios y sus consultas originales, y para cada $k \in \{2, \ldots, 50\}$ las correspondientes consultas protegidas de cada usuario. La Figura 3 muestra la utilidad de los datos desglosada según el parámetro k de la microagregación. Se puede observar como al crecer k (eso es, el número de usuarios por clúster crece), la pérdida de información del usuario también crece.

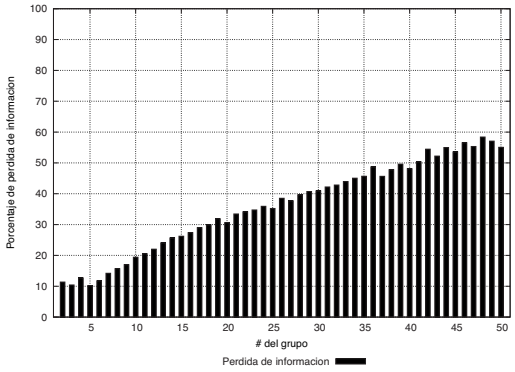


Figura 3: ILR medio para $k \in \{2, \ldots, 50\}$.

IV-C. Sobre la relación entre PPE y ILR

El PPE y el ILR nos ayudan a establecer un equilibrio entre privacidad y utilidad de los datos (pérdida de información).

Usando una k grande tenemos una gran pérdida de información (ILR), tal y como se puede observar en la Figura 3. Para $k = 35$ perdemos el $50\,\%$ de la información del usuario, y para $k = 3$ sólo perdemos un $10\,\%$. Por tanto, el mínimo ILR proporciona los datos más útiles, y consecuentemente debemos seleccionar los valores más pequeños de k.

Sin embargo, como se ve en la Figura 2, una k grande ofrece mayor privacidad para los usuarios, esto es, su perfil está menos expuesto. Por ejemplo, cuando $k = 2$, el $50\,\%$ del perfil del usuario está expuesto, y para $k = 10$ sólo el $10\,\%$ se expone. Desde el punto de vista de la privacidad, una k grande sería recomendable para proteger la privacidad de los usuarios. Sin embargo, consideramos que el perfil de un usuario está suficientemente protegido si su PPE es menor al $40\,\%$, es decir, $k = 3$ (ver [9]).

Así, podemos concluir que el valor óptimo de k para la microagregación es $k = 3$, porqué obtenemos un nivel de privacidad razonable (PPE) y una baja pérdida de información (ILR).

V. Conclusiones

En este artículo hemos introducido una técnica que garantiza el k-anonimato a nivel de usuario en registros de búsqueda mediante la microagregación. Para hacer público los registros, éstos tienen que ser protegidos para prevenir la revelación de información sensible así como la reidentificación de usuarios. Como siempre sucede en el campo de la SDC, existe un balance entre la privacidad y la utilidad. Mostramos como nuestra propuesta proporciona k-anonimato, manteniendo cierta información de los registros originales. Por consiguiente, nuestra propuesta se puede ver como un método eficiente y relativamente sencillo de proteger registros de búsqueda, si lo comparamos con otras propuestas existentes, y que permite garantizar, siempre que sea necesario, un alto nivel de anonimidad y privacidad.

Agradecimientos

Los autores agradecen las ayudas del MICINN (proyectos eAEGIS TSI2007-65406-C03-02, TSI2007-65406-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004), del Ministerio de Industria, Comercio y Turismo (proyecto TSI-020100-2009-720), y de el Gobierno de Cataluña (ayuda 2009 SGR 1135). G. Navarro-Arribas disfruta de una beca Juan de la Cierva (JCI-2008-3162) del MICINN.

Los autores son responsables de las ideas expresadas en este articulo, que no reflejan necesariamente la posición de la UNESCO ni comprometen a dicha organización.

Referencias

[1] E. Adar. User 4xxxxx9: Anonymizing query logs. In *Query Logs workshop*, 2007.

[2] M. Barbaro and T. Zeller. A face is exposed for AOL searcher no. 4417749. *The New York Times*, August 2006.

[3] A. Cooper. A survey of query log privacy-enhancing techniques from a policy perspective. *ACM Transactions on the Web*, 2(4), 2008.

[4] D. Defays and P. Nanopoulos. Panels of enterprises and confidentiality: the small aggregates method. In *Proc. of 92 Symposium on Design and Analysis of Longitudinal Surveys, Statistics Canada*, pages 195–204, 1993.

[5] J. Domingo-Ferrer and J. Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Engineering*, 14(1):189 – 201, 2002.

[6] J. Domingo-ferrer, J. Mateo-sanz, and V. Torra. Comparing sdc methods for microdata on the basis of information loss and disclosure. In *Proceedings of ETK-NTTS 2001, Luxemburg: Eurostat*, pages 807–826. Eurostat, 2001.

[7] J. Domingo-Ferrer and V. Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 11(2):195–212, September 2005.

[8] EFF. AOL's massive data leak. Electronic Frontier Foundation, http://w2.eff.org/Privacy/AOL/, 2009.

[9] A. Erola, J. Castellà-Roca, and A. Viejo. Exploiting social networks for improving privacy in personalized web search, 2010. *Submitted*.

[10] Y. Hong, X. He, J. Vaidya, N. Adam, and V. Atluri. Effective anonymization of query logs. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1465–1468, 2009.

[11] B.J. Jansen and A. Spink. An analysis of web searching by european alltheweb.com users. *Information Processing & Management*, 41(2):361–381, 2005.

[12] B.J. Jansen, A. Spink, and J. Pedersen. A temporal comparison of altavista web searching: Research articles. *Journal of the American Society for Information Science and Technology*, 56(6):559–570, 2005.

[13] B.J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing & Management*, 36(2):207 – 227, 2000.

[14] Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. Releasing search queries and clicks privately. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 171–180, 2009.

[15] R. Kumar, J. Novak, B. Pang, and A. Tomkins. On anonymizing query logs via token-based hashing. In *16 International World Wide Web Conference*, pages 629–638, 2007.

[16] Wang Lixia and Han Jianmin. Utility evaluation of k-anonymous data by microaggregation. In *International Conference on Communication System, Networks and Applications, 2009 ICCSNA*, volume 4, pages 381 –384, 2009.

[17] E. Mills. AOL sued over web search data release. CNET News, http://news.cnet.com/8301-10784_3-6119218-7.html, September 2006.

[18] G. Navarro-Arribas and V. Torra. Tree-based microaggregation for the anonymization of search logs. In *WI-IAT '09: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 155–158, 2009.

[19] A. Oganian and J. Domingo-Ferrer. On the complexity of optimal microaggregation for statistical disclosure control. *Statistical Journal of the United Nations Economic Commision for Europe*, 18(4):345–353, 2001.

[20] B. Poblete, M. Spiliopoulou, and R. Baeza-Yates. Website privacy preservation for query log publishing. In *First International Workshop on Privacy, Security, and Trust in KDD (PinKDD 2007)*, pages 80–96, 2008.

[21] P. Samarati. Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[22] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5), 2002.

[23] V. Torra. Microaggregation for categorical variables: A median based approach. In *Proc. Privacy in Statistical Databases (PSD 2004)*, volume 3050 of *LNCS*, pages 162–174, June 2004.

[24] V. Torra. Constrained microaggregation: Adding constraints for data editing. *Transactions on Data Privacy*, 1(2):86–104, 2008.

[25] L. Xiong and E. Agichtein. Towards privacy-preserving query log publishing. In *Query Log Analysis: Social and Technological Challenges, Workshop in 16 International World Wide Web Conference*, 2007.

El Juego de Recuperación de Información con Privacidad de Usuario Por Pares

Josep Domingo-Ferrer
Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
Càtedra UNESCO de Privadesa de Dades
Email: josep.domingo@urv.cat

Úrsula González-Nicolás
Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
Càtedra UNESCO de Privadesa de Dades
Email: ursula.gonzaleznicolas@urv.cat

***Resumen*—En la recuperación de información con privacidad (RIP), el usuario quiere obtener información de una base de datos sin que ésta conozca lo que está buscando. La mayoría de los protocolos actuales sobre RIP son poco apropiados para motores de búsqueda o grandes bases de datos, debido a su complejidad y a la suposición de que la base de datos coopera activamente en el protocolo de RIP. Con una base de datos no-cooperativa, puede ser útil una comunidad por pares (*peer-to-peer*, P2P), donde los pares sometan consultas por cuenta de un usuario. De esta forma se obtiene la recuperación de información con privacidad de usuario (RIPU), una propiedad mediante la cual la base de datos conoce la información que se ha recuperado pero sin saber por quién.**

En este artículo se presenta un análisis mediante teoría de juegos de RIPU por pares (RIPUP): i) se especifica una métrica para medir la privacidad de un par frente a la base de datos y frente al resto de los pares; ii) se calcula la utilidad de las diferentes estrategias que puede seguir un par con el objetivo de maximizar su privacidad; iii) mediante la maximización de esta privacidad, para cada nueva consulta se obtiene un comportamiento racional a seguir para cada par, lo que permite la automatización del proceso de decisión de los pares.

Una conclusión relevante es que un par debe ayudar al resto de pares con el objetivo de maximizar su propia utilidad de privacidad; también es destacable que, cuanto más heterogénea sea la tasa de generación de consultas por parte de los pares, más racionalmente serviciales serán estos.

***Palabras clave*— Recuperación de información con privacidad (*Private information retrieval*), Recuperación de información con privacidad de usuario (*User-private information retrieval*), Preservación de la privacidad en minería de datos (*Privacy-preserving data mining*), Teoría de juegos (*Game theory*).**

I. Introducción

El objetivo de la recuperación de información con privacidad (RIP) es permitir a un usuario obtener información de una base de datos sin que ésta sepa en qué información está interesado. En la literatura sobre RIP (véanse los artículos seminales [6], [7] y los más recientes [11], [1]) la base de datos se modela como un vector, y se supone que un usuario desea recuperar el i-ésimo elemento del vector manteniendo el índice i oculto ante la base de datos. Si bien esta visión de los protocolos de RIP permite desarrollos teóricos, existen algunas suposiciones que dificultan su implementación en la práctica:

- En general, la base de datos no puede modelarse como un vector en el que el usuario conoce la dirección física i del campo en el que está interesado (*e.g.* piénsese en un usuario haciendo una consulta a un motor de búsqueda);
- Si la base de datos contiene n elementos, los protocolos teóricos de RIP tienen una complejidad $O(n)$ [6], [7]: el protocolo debe "tocar" todos los registros para evitar dar al servidor cualquier pista sobre el valor de i, lo que no es viable para grandes bases de datos y/o motores de búsqueda [3];
- Se supone que el servidor de la base de datos coopera en el protocolo de RIP; es el usuario quien está interesado en proteger su propia privacidad, mientras que la motivación para el servidor de la base de datos es dudosa; en realidad, es probable que la RIP sea poco atractiva para la mayoría de las compañías que ofrecen bases de datos consultables, ya que limita su capacidad de generación de perfiles.

Por las razones anteriores, en la práctica se deben flexibilizar las suposiciones de RIP. A continuación repasamos algunas propuestas.

En primer lugar, se debe tener en cuenta que los sistemas de *onion-routing* como Tor [12] no están pensados para ofrecer recuperación de información con privacidad. Estos sistemas protegen el transporte de datos, pero no ofrecen ninguna protección extremo a extremo (a nivel de aplicación). Como un motor de búsqueda o un servidor de base de datos puede vincular las consultas consecutivas sometidas por el mismo usuario (*e.g* mediante el uso de galletas), se puede crear un perfil y reidentificar al usuario.

En [5] se propone un sistema denominado Goopir, en el que el usuario enmascara su consulta añadiendo $k-1$ consultas falsas y luego envía la consulta enmascarada resultante a un motor de búsqueda o a una base de datos grande, que no tiene por qué cooperar (de hecho, ni siquiera tiene que saber que el usuario está tratando de proteger su privacidad). Estrictamente hablando, Goopir no da RIP según lo definido anteriormente, sino que proporciona $h(k)$-RIP, ya que encubre la consulta dentro de un conjunto de k consultas de entropía al menos $h(k)$. Este sistema funciona bien, pero supone que las frecuencias de las palabras clave y expresiones que pueden aparecer en una consulta son conocidas y están disponibles:

para una mayor privacidad, las frecuencias de la consulta real y las falsas deben ser similares, por lo que la incertidumbre $h(k)$ del motor de búsqueda sobre la consulta real es máxima.

TrackMeNot [13] es otro sistema práctico basado en un principio diferente: en lugar de someter una única consulta enmascarada por cada consulta real como hace Goopir, se instala una extensión en el navegador del computador del usuario para ocultar las consultas reales de éste en un conjunto de consultas "fantasma" automáticas sometidas a los motores de búsqueda más populares en diferentes intervalos de tiempo. Si bien esto es factible a pequeña escala, si el uso de TrackMeNot se generalizara, la sobrecarga introducida por las consultas fantasma degradaría de forma significativa el rendimiento de los motores de búsqueda y las redes de comunicaciones. Por otra parte, el intervalo de sumisión de las consultas fantasma automáticas puede ser distinguible respecto al intervalo de sumisión de las consultas reales, lo que daría pistas a un intruso para identificar las consultas reales.

Con el mismo espíritu práctico, el sistema descrito en [8] encubre a un usuario en una comunidad anónima de pares (*peer-to-peer*, P2P). El usuario somete consultas en nombre de los pares anónimos y viceversa. Las parejas de usuarios de la comunidad P2P comparten claves de cifrado simétricas que utilizan para establecer un canal confidencial. De este modo, la base de datos sigue conociendo qué elemento se quiere recuperar (lo que se desvía de la RIP estricta), pero no puede obtener los historiales de las consultas reales de los usuarios, que quedan difusos entre los usuarios pares; llamamos a esta flexibilización de RIP recuperación de información con privacidad de usuario (RIPU). Este enfoque tiene sin duda algunas ventajas: a diferencia de [5], no requiere ningún conocimiento de las frecuencias de todas las posibles palabras clave y expresiones a consultar; a diferencia de [13] evita la sobrecarga de la sumisión de consultas fantasma. Además de preservar la privacidad del perfil de consultas del usuario frente a la base de datos y a intrusos externos, el sistema en [8] ofrece privacidad frente a los usuarios pares, porque los pares son anónimos entre sí.

En [4] se propone otro enfoque RIPUP. Aquí se logra el anonimato con la ayuda de un nodo central que pone en contacto a un grupo de n usuarios que quieren someter una consulta. Los usuarios ejecutan un protocolo de recuperación de consultas anónimas por el que cada usuario puede obtener una consulta de uno de los $n-1$ usuarios restantes, sin saber a qué usuario pertenece la consulta. Cada usuario envía la consulta recibida a la base de datos, y retransmite al resto de usuarios la respuesta obtenida de la base de datos. De este modo, el usuario que originalmente publicó la consulta puede obtener la respuesta correspondiente.

En [14] se presenta una alternativa RIPUP basada en una red social. Básicamente, un usuario A emite una consulta hacia uno de sus contactos en la red social, llamado B; B o bien somete la consulta directamente a la base de datos o bien la envía a uno de sus contactos, por ejemplo C, y así sucesivamente hasta que un nodo someta la consulta. La respuesta de la consulta de A es devuelta a través del camino inverso. El objetivo es mantener las consultas de cada usuario A frente al resto de la forma más uniforme posible, obteniendo así la máxima privacidad frente a la base de datos. Una característica atractiva de este sistema es que se utiliza una red social existente como una comunidad de pares, lo que hace que la implementación sea más fácil.

I-A. Contribución y plan del artículo

Los sistemas RIPUP como los propuestos en [8] y [14] se pueden formalizar como un juego. En este artículo se especifica una métrica para la privacidad de los pares frente a la base de datos. Se calculan las utilidades de privacidad de las diferentes estrategias que puede seguir un par con el objetivo de maximizar su privacidad. Maximizar la utilidad de privacidad para cada nueva consulta produce un comportamiento racional para cada par, lo que permite automatizar el proceso de decisión de estos. En particular, se determinan las condiciones bajo las que se alcanza un equilibrio de Nash [9], [10] entre pares. Dado este equilibrio, la mejor opción para que un par maximice su privacidad es que otro par someta su consulta, y la mejor opción para que el resto de pares maximice su privacidad es someter la consulta de otro par a la base de datos.

El juego RIPUP se formaliza en la Sección II. Los resultados de simulación se presentan en la Sección III. La Sección IV resume las conclusiones y las futuras líneas de investigación.

II. Formalización del juego RIPUP

Consideremos ahora un sistema con N jugadores Q^1, $\cdots$, Q^N, y un servidor de base de datos BD. Para cada i, supongamos que Q^i origina una consulta. Luego Q^i tiene dos estrategias posibles a seguir:

- Sii: Q^i somete su consulta directamente a la BD;
- Sij: Q^i envía su consulta a Q^j, para alguna $j \neq i$ y solicita a Q^j que someta la consulta en nombre de Q^i.

Cuando Q^j recibe la consulta de Q^i, éste tiene dos posibles estrategias a seguir:

- Tji: Q^j somete la consulta de Q^i a la BD y devuelve el resultado a Q^i;
- Tjj: Q^j ignora la consulta de Q^i y no hace nada.

Sea $X^i(t) = \{x^i_1, \cdots, x^i_{m_i(t)}\}$ el conjunto de consultas originadas por Q^i hasta el instante t. Sea $Y^i(t) = \{y^i_1, \cdots, y^i_{n_i(t)}\}$ el conjunto de consultas sometidas por Q^i a la BD hasta el instante t. Sea $Y^{ij}(t) \subseteq X^i(t)$ el conjunto de consultas originadas por Q^i y enviadas a Q^j hasta el instante t. Definimos el estado $\Lambda^i(t)$ de Q^i en el tiempo t como las siguientes $N+1$ tuplas

$$(o^i(x^i_1, t), \cdots, o^i(x^i_{m_i(t)}, t))$$

$$(s^i(y^i_1, t), \cdots, s^i(y^i_{n_i(t)}, t))$$

$$(f^{ij}(x^i_1, t), \cdots, f^{ij}(x^i_{m_i(t)}, t))$$

para $j \in \{1, \cdots, N\}$, con $j \neq i$, donde $o^i(x^i_k, t)$ es el número de veces que la consulta $x^i_k \in X^i(t)$ ha sido originada por

Q^i hasta el instante t, $s^i(y_k^i, t)$ es el número de veces que la consulta $y_k^i \in Y^i(t)$ ha sido sometida por Q^i, y $f^{ij}(x_k^i, t)$ es el número de veces que la consulta x_k^i orginada por Q^i ha sido enviada a Q^j hasta el instante de tiempo t.

Claramente,

$$\sum_{j \in \{1,\cdots,N\}\setminus\{i\}} f^{ij}(x_k^i, t) \leq o(x_k^i, t)$$

Definimos $d^i(t) = |X^i(t) \cup Y^i(t)|$ y

$$X^i(t) \cup Y^i(t) = \{z_1^i, \cdots, z_{d^i(t)}^i\}$$

Para $l = 1, \cdots, d^i(t)$, definimos

$$o^i(z_l^i, t) = \begin{cases} o(x_k^i, t) \text{ if } z_l^i = x_k^i \in X^i(t) \text{ para cualquier } k \\ 0 \text{ if } z_l^i \notin X^i(t) \end{cases}$$

$$s^i(z_l^i, t) = \begin{cases} s(y_k^i, t) \text{ if } z_l^i = y_k^i \in Y^i(t) \text{ para cualquier } k \\ 0 \text{ if } z_l^i \notin Y^i(t) \end{cases}$$

$$f^{ij}(z_l^i, t) = \begin{cases} f(x_k^i, t) \text{ if } z_l^i = x_k^i \in X^i(t) \text{ para cualquier } k \\ 0 \text{ if } z_l^i \notin X^i(t) \end{cases}$$

La privacidad de Q^i frente a la BD en el instante t se define como

$$d(X^i(t), Y^i(t)) = \sqrt{\sum_{l=1}^{d^i(t)} (o^i(z_l^i, t) - s^i(z_l^i, t))^2}$$

La privacidad de Q^i frente a Q^j en el instante t se define como

$$d(X^i(t), Y^{ij}(t)) = \sqrt{\sum_{l=1}^{d^i(t)} (o^i(z_l^i, t) - f^{ij}(z_l^i, t))^2}$$

Las utilidades de las estrategias de Q^i y Q^j son

$$U_{Sii}(t+1) = (d(X^i(t+1), Y^i(t+1)))^{\alpha_{i,BD}} \prod_{r \in \{1,\cdots,N\}\setminus\{i\}} (d(X^i(t+1), Y^{ir}(t)))^{\alpha_{i,r}}$$

y

$$U_{Sij}(t+1) = (d(X^i(t+1), Y^i(t)))^{\alpha_{i,BD}} \prod_{r \in \{1,\cdots,N\}\setminus\{i\}} (d(X^i(t+1), Y^{ir}(t+1)))^{\alpha_{i,r}}$$

donde $\alpha_{i,BD}$ y $\alpha_{i,r}$ son los pesos en el intervalo $[0, 1]$ que indican la importancia de la privacidad de Q^i frente a la BD y Q^r, respectivamente. Con Tji, tenemos $X^j(t+1) = X^j(t)$, y $Y^{jr}(t+1) = Y^{jr}(t)$ para todos los pares $r \neq j$; por lo tanto:

$$U_{Tji}(t+1) = (d(X^j(t), Y^j(t+1)))^{\alpha_{j,BD}} \prod_{r \in \{1,\cdots,N\}\setminus\{j\}} (d(X^j(t), Y^{jr}(t)))^{\alpha_{j,r}}$$

Finalmente, puesto que Q^j no realiza ninguna acción en Tjj, tenemos que

$$U_{Tjj}(t+1) = U_{Tjj}(t) = (d(X^j(t), Y^j(t)))^{\alpha_{j,BD}} \prod_{r \in \{1,\cdots,N\}\setminus\{j\}} (d(X^j(t), Y^{jr}(t)))^{\alpha_{j,r}}$$

Dependiendo del estado de Q^i en el instante t para la consulta x_k^i originada en el instante $t+1$, Sii o Sij para alguna j puede ser la estrategia que proporcione mayor utilidad a Q^i; lo mismo vale para las estrategias Tjj y Tji para cualquier par Q^j. Se debe tener en cuenta que, si $U_{Tji}(t+1) \leq U_{Tjj}(t+1)$ para todo Q^j, la mejor estrategia para Q^i es Sii. Dado que Q^i no conoce las utilidades del resto de los pares, Q^i debe utilizar el procedimiento de ensayo-error descrito en el Algoritmo 1 para encontrar el mejor par.

Algorithm 1 BÚSQUEDA DEL MEJOR PAR PARA Q^i

```
ND = {Q^1, ..., Q^N}{Todos los pares son candidatos}
sometido = 0
while sometido = 0 do
  Calcular Q^{j*} = arg máx_{Q^j ∈ ND} U_{Sij}(t+1)
  Sea Q^{j*} el candidato al mejor par
  if j* = i then
    Someter x_k^i {Q^i somete su propia consulta}
    s^i(x_k^i, t+1) := s^i(x_k^i, t) + 1
    sometido = 1
  else
    Enviar x_k^i a Q^{j*};
    f^{ij*}(x_k^i, t+1) := f^{ij*}(x_k^i, t) + 1;
    if Q^{j*} rechaza someter x_k^i then
      Excluir Q^{j*} de ND {Q^{j*} rechaza la sumisión si
      U_{Tj*i}(t+1) ≤ U_{Tj*j*}(t+1)}
    else
      sometido = 1
    end if
  end if
end while
```

Obsérvese que cada intento fallido en el Algoritmo 1 implica una cierta pérdida de privacidad, ya que la consulta originada se revela a cada par candidato. Ello es debido a la actualización de $f^{ij*}(x_k^i, t+1)$, que influye en el cálculo del siguiente candidato Q^{j*}.

Proposición 1: Si el mejor par es $Q^{j*} \neq Q^i$ y $U_{Sii}(t+1) < U_{Sij*}(t+1)$, entonces $(Sij*(t+1), Tj*i(t+1))$ es un equilibrio de Nash para el juego entre Q^i y Q^{j*} con estados$(\Lambda^i(t), \Lambda^{j*}(t))$.

Demostración: Si $Q^{j*} \neq Q^i$, se tiene que $U_{Sii}(t+1) \leq U_{Sij*}(t+1)$ y $U_{Tj*i}(t+1) > U_{Tj*j*}(t+1)$, y la proposición se cumple. □

III. RESULTADOS EXPERIMENTALES

En esta sección, presentamos los resultados experimentales. En las simulaciones, el Algoritmo 1 encuentra el mejor par casi siempre en el primer intento, por lo que podemos decir que es computacionalmente barato.

El tiempo transcurrido entre dos consultas consecutivas de un par Q^i se ha generado muestreando una variable aleatoria con distribución exponencial con parámetro λ_i. Por lo tanto, la tasa de generación de consultas de Q^i es de λ_i consultas por unidad de tiempo, y el tiempo de espera entre consultas sucesivas es de $1/\lambda_i$.

III-A. *Métricas de colaboración*

Las métricas consideradas para analizar la (falta de) colaboración entre pares son:

- *Consultas Sometida Directamente (CSD).* Dado un par Q^i, $CSD^i(t)$ es la fracción de las consultas generadas por Q^i hasta el instante de tiempo t que han sido sometidas de forma directa a la BD por Q^i.
- *Consultas Rechazadas (CR).* Dado un par Q^i, $CR^i(t)$ es la fracción de las consultas recibidas por Q^i del resto de pares hasta el instante de tiempo t, que Q^i ha rechazado a someter.

III-B. *Influencia de la tasa de generación de consultas*

Se ha simulado un juego de N pares con $N = 5$ durante 550 unidades de tiempo. Suponemos que todos los pares se asignan la misma importancia de privacidad unos a otros y a la BD, y consideramos varias tasas de generación de consultas. Los valores de las consultas se han elegido al azar uniformemente entre un conjunto de 1000 consultas disponibles. La Figura 1 muestra la evolución de las métricas CSD y CR cuando los cinco pares tienen la misma tasa de generación de consultas $\lambda = 1$. Se puede observar que CSD se estabiliza entre el 50 % y el 70 %, mientras que CR se estabiliza entre el 0 % y el 30 %.

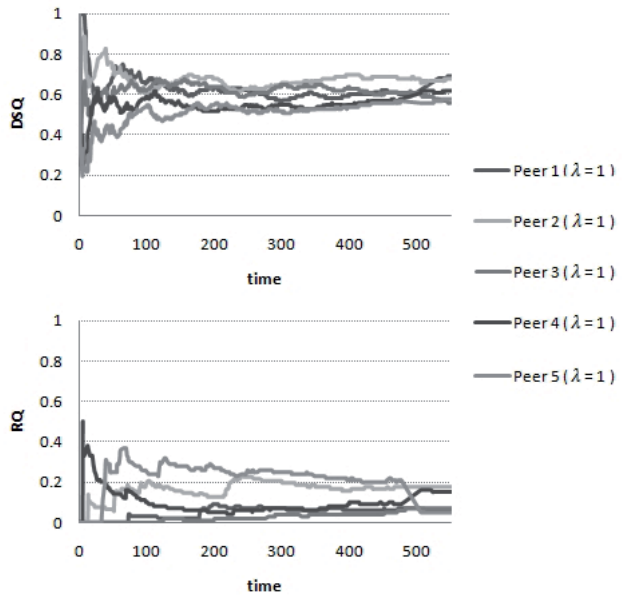


Figura 1: Evolución de CSD (arriba) y CR (abajo) de cinco pares con la misma tasa de generación de consultas ($\lambda = 1$) y las mismas importancias de privacidad asignadas a 1

La Figura 2 muestra la evolución de CSD y CR cuando los cinco pares tienen tasas de generación de consultas distintas: $\lambda_1 = 1$, $\lambda_2 = 0{,}8$, $\lambda_3 = 0{,}6$, $\lambda_4 = 0{,}4$ y $\lambda_5 = 0{,}2$. Un par con una tasa de generación de consultas comparativamente mayor, es decir, con una λ_i mayor, tiende a tener un $CSD^i(t)$ comparativamente menor. Esto ocurre debido a que el par perdería una cantidad considerable de privacidad frente a la BD si éste se sometiera directamente todas sus (frecuentes) consultas. Para evitar esta pérdida, una mayor proporción de sus consultas debe ser sometida por otros pares. El comportamiento de CR es muy interesante y diferente del que se muestra en la Figura 1: en la Figura 2 la fracción CR de consultas rechazadas es mucho más baja para todos los pares. La explicación es que, como en la simulación anterior los pares generaban más consultas, tenían una mayor necesidad de someter consultas de otros pares con el fin de difuminar su propio perfil de consultas frente a la BD; en este caso en que los pares generan menos consultas, les favorece someter consultas de otros pares, ya que les ayudan a ocultar sus pocas consultas.

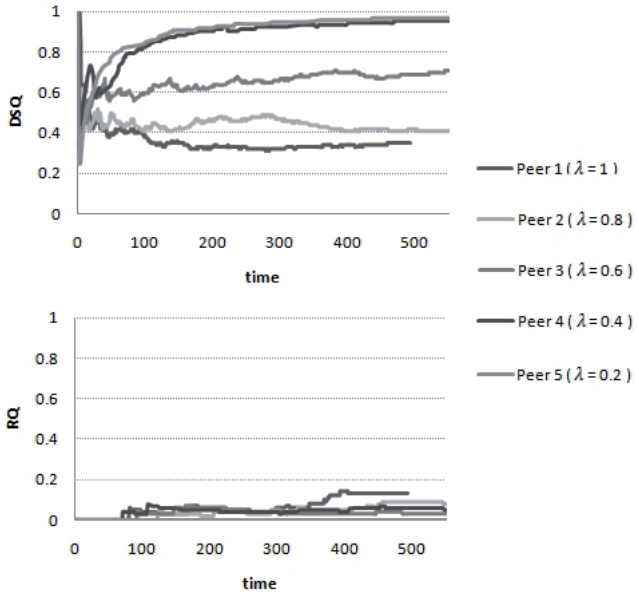


Figura 2: Evolución de CSD (arriba) y CR (abajo) de cinco pares con tasas de generación de consultas diferentes y las mismas importancias de privacidad asignadas a 1

III-C. *Resultados utilizando consultas reales*

Para estudiar el sistema en un entorno real, se ha utilizado el volcado de una base de datos de AOL [2], con veinte millones de consultas procedentes del estudio de las consultas de 650000 usuarios durante un período de 3 meses. Para realizar esta prueba hemos elegido al azar cinco usuarios de entre los que tenían un mínimo de 600 consultas, con sus respectivas consultas emitidas por primera vez el 1 de Marzo de 2006. Estos usuarios emitieron sus consultas posteriores con intervalos muy diferentes entre una consulta y la siguiente, hasta el 31 de Mayo de 2006. Hemos tomado estos cinco usuarios, con los valores y los tiempos de sus consultas como si se correspondieran a cinco usuarios en nuestro sistema.

La Figura 3 muestra la evolución de CSD y CR de estos cinco usuarios AOL. El gráfico se detiene tan pronto como uno de los usuarios ha alcanzado su consulta número 600, lo que sucedió por primera vez para el Usuario 4. Por lo tanto, las abscisas son proporcionales al número de consultas generadas por el Usuario 4 en lugar del tiempo. La Figura 3

se asemeja a la Figura 2 hasta cierto punto. La evolución de CSD es diferente para cada usuario: los usuarios con una tasa de generación de consultas más alta tienen un menor CSD. Por otra parte, CR es cero para todos los usuarios durante todo el período de tiempo: todo el mundo está siempre dispuesto a ayudar a otros usuarios en la sumisión de sus consultas a la BD.

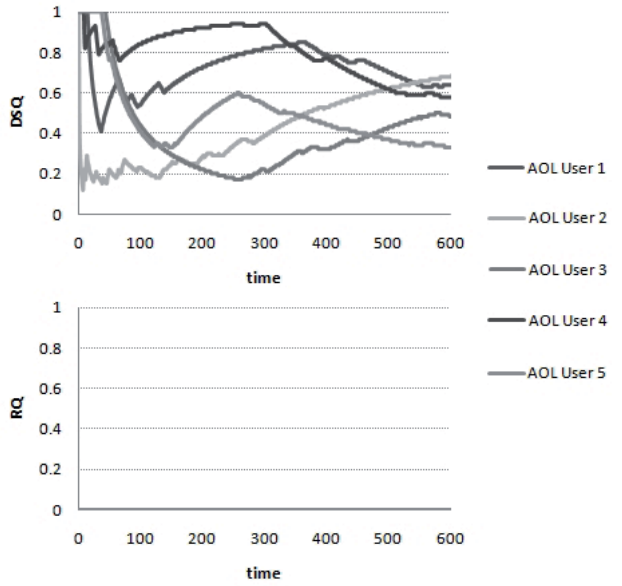


Figura 3: Evolución de CSD (arriba) y CR (abajo) de cinco usuarios-pares correspondientes a usuarios reales de la base de datos de AOL

IV. Conclusiones y trabajo futuro

En este artículo se ha especificado una métrica de privacidad basada en distancia para pares frente a una base de datos y frente a otros pares en un sistema RIPUP. A partir de esta métrica, se ha calculado la utilidad de privacidad de cada par. Cuando un par genera una nueva consulta, éste y el resto de pares toman una serie de decisiones racionales para maximizar su privacidad. Curiosamente, estas decisiones racionales conducen a menudo a los pares a ayudarse en la sumisión de consultas a la base de datos. De este modo, ser egoístamente racional en términos de privacidad lleva a ayudar a otros pares. En particular, se describen las condiciones en las que existe un equilibrio de Nash entre pares; en este equilibrio, la mejor opción para que un par maximice su privacidad es que otro par someta su consulta, y la mejor opción para que el resto de pares maximice su privacidad es someter la consulta de otro par a la base de datos. Un resultado empírico muy interesante es que, cuando la tasa de generación de consultas de los pares es heterogénea, estos están más dispuestos a ayudarse unos a otros en la sumisión de sus consultas: la frecuencia de consultas hace que sea conveniente enviar consultas de otros pares para esconder las propias, y esto también sucede cuando la frecuencia de consultas es baja. Por alguna razón, cuanto más se desvíe la tasa generación de consultas de la homogeneidad, mayor será la puesta en peligro de la privacidad de los pares y más se necesitaran entre ellos.

El modelo propuesto para un comportamiento racional automatizado en los sistemas RIPUP se puede ampliar de varias maneras:

- Generalizarlo a RIPUP multi-salto, en el que un par pueda decidir someter la consulta de otro a la base de datos o enviarla a un tercero, y así sucesivamente.
- Estudiar otras métricas de privacidad que no sean las basadas en distancias consideradas en este artículo, como por ejemplo información mutua.
- Además de privacidad, incluir otras propiedades funcionales en las utilidades de los pares (*e.g.* el tiempo de respuesta de la consulta, número de saltos, etc.)

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por la Generalitat de Catalunya mediante la subvención 2009 SGR 1135 y por el MICINN a través de los proyectos TSI2007-65406-C03-01 “E-AEGIS” y CONSOLIDER INGENIO 2010 CSD2007-00004 “ARES”. El primer autor recibe financiación de la Generalitat de Catalunya como investigador ICREA-Acadèmia. Los autores forman parte de la Cátedra UNESCO de Privacidad de Datos, pero sus opiniones no necesariamente reflejan la posición de la UNESCO ni comprometen a dicha organización.

Referencias

[1] C. Aguilar-Melchor and Y. Deswarte, “Trustable relays for anonymous communication”, *Transactions on Data Privacy*, 2(2):101-130, 2009.

[2] “AOL Search Data Scandal”, August 2006. http://en.wikipedia.org/wiki/AOL_search_data_scandal

[3] A. Beimel, Y. Ishai, and T. Malkin, “Reducing the servers” computation in private information retrieval: Pir with preprocessing”, *Journal of Cryptology*, 17:125–151, 2004.

[4] J. Castellà-Roca, A. Viejo and J. Herrera-Joancomartí, “Preserving user’s privacy in web search engines”, *Computer Communications*, 32(13-14):1541-1551, 2009.

[5] J. Domingo-Ferrer, A. Solanas and J. Castellà-Roca, “$h(k)$-Private information retrieval from privacy-uncooperative queryable databases”, *Online Information Review*, 33(4):720–744, 2009.

[6] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, “Private information retrieval”, in *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 41–50, 1995.

[7] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, “Private information retrieval”, *Journal of the ACM*, 45:965–981, 1998.

[8] J. Domingo-Ferrer, M. Bras-Amorós, Q. Wu and J. Manjón, “User-private information retrieval based on a peer-to-peer community”, *Data and Knowledge Engineering*, 68(11):1237–1252, 2009.

[9] J. Nash, “Non-cooperative games”, *Annals of Mathematics*, 54:289-295, 1951.

[10] N. Nisan, T. Roughgarden, É. Tardos and V. V. Vazirani (eds.), *Algorithmic Game Theory*. Cambridge University Press, 2007.

[11] R. Ostrovsky and W. E. Skeith-III, “A survey of single-database PIR: techniques and applications”, in *Public Key Cryptography-PKC 2007*, volume 4450 of *Lecture Notes in Computer Science*, pages 393–411, Berlin Heidelberg, 2007.

[12] The Tor Project, Inc, “Tor: Overview”, http://torproject.org/overview.html.en

[13] D. C. Howe and H. Nissenbaum, “TrackMeNot: Resisting surveillance in web search”, in *Lessons from the Identity Trail: Privacy, Anonymity and Identity in a Networked Society*. I. Kerr, C. Lucock and V. Steeves(eds.). Oxford University Press, Oxford UK, 2009. Software downloadable from:http://www.mrl.nyu.edu/~dhowe/trackmenot/

[14] A. Viejo and J. Castellà-Roca, “Using social networks to distort users’ profiles generated by web search engines”, *Computer Networks* (to appear).

Técnicas de Anonimato para Securizar Redes Móviles Ad Hoc

Oscar Manso
Internet Interdisciplinary Institute
Universitat Oberta de Catalunya
Email: omanso@uoc.edu

Helena Rifà-Pous
Internet Interdisciplinary Institute
Universitat Oberta de Catalunya
Email: hrifa@uoc.edu

***Resumen*—Las redes móviles ad hoc son redes formadas por la interconexión de terminales inalámbricos que de manera autónoma, sin ninguna administración central, establecen enlaces de comunicación entre ellos. La infraestructura de red la componen los propios terminales de usuarios que actúan de gestores y encaminadores de paquetes. Así, un usuario cualquiera puede conectarse con un terminal remoto a través de una conexión multisalto entre diferentes usuarios. En este tipo de redes tan abiertas, uno de los retos prioritarios es proteger el anonimato de los sujetos y sus localizaciones. En este artículo hacemos un repaso de las técnicas existentes a través de los protocolos que se han propuesto en la literatura, y exponemos los problemas que aun quedan abiertos.**

I. Introducción

La proliferación de dispositivos móviles en el mercado ha hecho surgir nuevos medios de comunicación basados en la creación de redes formadas por terminales de usuarios que se agrupan entre sí de forma esporádica y permiten construir una base de comunicación. Este tipo de redes reciben el nombre de redes móviles ad hoc (MANET) y están tomando especial importancia por ser unas redes que no precisan de infraestructura dedicada, son rápidas de desplegar, pueden proporcionar acceso a la información en entornos aislados y/o conflictivos, y su coste es reducido.

La transmisión de datos en las redes ad hoc se realiza a través de los propios terminales de los usuarios, que actúan como encaminadores de los paquetes. Así, el rango de comunicación de un usuario se extiende más allá del alcance de sus radiaciones electromagnéticas, pudiendo crear enlaces de comunicaciones con nodos remotos. Empero, el hecho de transmitir los datos a través de terminales finales entraña una clara amenaza a la privacidad de los usuarios.

Proporcionar servicios de comunicación anónima es una propiedad muy deseable en redes MANET. Sin embargo la arquitectura particular de las MANET conlleva una serie de características funcionales que son únicas y específicas de este tipo de redes, y que impiden la adopción directa de los esquemas de anonimato diseñados por redes cableadas. En concreto, cabe destacar las siguientes características:

- La autogestión de la red se realiza a través de un medio abierto y susceptible a ataques externos, tanto activos como pasivos. Los enlaces de comunicaciones inalámbricos permiten que los mensajes sean escuchados fácilmente por usuarios que no son sus legítimos receptores.
- La capacidad de la red y la responsabilidad de que ésta funcione está distribuida entre todos sus miembros. Los dispositivos que forman la red son terminales genéricos, con una baja protección física. La probabilidad que algunos de estos terminales sean comprometidos no es irrelevante. Ello implica tener buenos sistemas de detección de intrusiones y gestión de la confianza que permitan proteger a la red de ataques internos.
- La topología de red es dinámica y caótica. En general, los terminales de la red son móviles y su estado en la red es transitorio e irregular. Uno de los principales desafíos en este tipo de entornos es el descubrimiento y mantenimiento de rutas de manera eficiente y anónima.
- Los recursos de la red son limitados. Los terminales tienen una capacidad de proceso, memoria y batería reducidos. El ancho de banda también es limitado, y al utilizar redes inalámbricas los canales de transmisión sufren interferencias y devaneos.

En este artículo definiremos las propiedades que son necesarias para proporcionar comunicaciones anónimas en redes ad hoc, y revisaremos el estado del arte de las técnicas de anonimato propuestas así como las funcionalidades que implementan los diferentes protocolos para MANETs. El resto del artículo está organizado de la siguiente forma. En la sección II introducimos y clasificamos las propiedades de anonimato. A continuación revisamos las soluciones propuestas de anonimato de usuarios (sección III), desvinculación del origen de los mensajes (sección IV), y indetectabilidad de la actividad (sección V). Finalmente, la sección VI presenta los principales problemas abiertos del anonimato en MANETs.

II. Anonimato

En primer lugar definimos las propiedades relacionadas con una comunicación anónima:

1. **Anonimato de los sujetos**: propiedad de no ser identificable entre un conjunto de sujetos.
2. **Desvinculación de mensajes**: propiedad de ocultar la relación que hay entre una comunicación y los sujetos que la llevan a cabo.
3. **Indetectabilidad**: incapacidad de distinguir si un elemento existe o no. Si consideramos mensajes, la indetectabilidad supondría que éstos no son suficientemente discernibles de, por ejemplo, ruido blanco.

Las propiedades de anonimato de una red pueden ser violadas por diferentes ataques, tanto activos como pasivos. En los ataques activos los usuarios maliciosos participan en el protocolo de red al que pretenden quebrantar, ya sea a través de ataques externos (como usuarios ajenos al sistema) o ataques internos (como miembros lícitos de la red). Por otro lado, los ataques pasivos no perturban el normal funcionamiento de los protocolos de red, los atacantes escuchan de forma no-autorizada los paquetes que se transmiten por la red y a través de un análisis de tráfico extrapolan información como las rutas de transmisión, el contenido de los mensajes, o la identidad, posición o movimiento de los nodos.

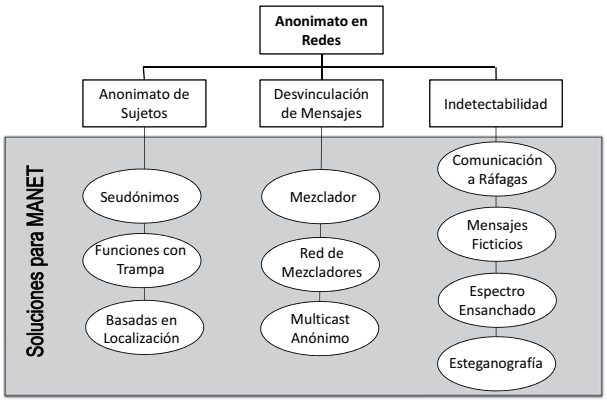


Figura 1. Taxonomía de las técnicas de anonimato para MANETs

En la figura 1 mostramos una taxonomía de las técnicas de anonimato para MANETs: los rectángulos simbolizan las propiedades de anonimato, y las elipses el tipo de soluciones que se han propuesto en la literatura. En los siguientes apartados analizaremos con más detalle las propiedades de anonimato y las soluciones propuestas. La tabla I hace un resumen de las principales soluciones adoptadas por los diferentes protocolos de anonimato en MANET.

	Anonimato sujetos			Desvincul. mensajes		Indetectabilidad
Protocolo	Sinon.	FTramp.	Loc.	Casc.	RLibre	MFict.
R-AO2P [20]	S	-	S	S	-	-
MASK [23]	S	-	-	-	S	Destino
ANODR [14]	S	S	-	S	-	-
ASR [24]	S	S	-	S	-	S
ANONDSR [18]	-	S	-	S	-	S
D-ANODR [21]	S	-	-	S	-	-
ARM [17]	S	S	-	S	-	S
ARMR [11]	S	S	-	-	S	S
ODAR [19]	S	-	-	S	-	-
SDAR [3]	-	S	-	S	-	-
ALARM [8]	S	-	S	S	-	-
PRISM [9]	-	-	S	S	-	-
ANAP [6]	S	S	-	S	-	S
RIOMO [15]	S	-	-	S	-	S
SDDR [12]	-	S	-	S	-	-

Cuadro I
CARACTERÍSTICAS DE LOS PROTOCOLOS DE ANONIMATO PARA MANET

III. ANONIMATO DE LOS SUJETOS

En este apartado se introducen las tecnologías que permiten proporcionar anonimato a los sujetos. Cada una de ellas acomete unos usos específicos de la red.

- **Seudónimos**: Usados como identificadores anónimos de los nodos de un vecindario. Facilitan las comunicaciones salto-a-salto, y que sobre éstas se construyan rutas multisalto.
- **Funciones unidireccionales con trampa** (*Trapdoor functions*): Permiten la búsqueda de nodos anónimos remotos. Se usan en la fase de descubrimiento de rutas de las MANET.
- **Rutas basadas en la localización**: Permite crear rutas anónimas entre nodos situados en puntos concretos de la red.

III-A. Seudónimos

Una forma de esconder la identidad de los sujetos que actúan en una comunicación es a través de seudónimos. Éstos fueron introducidos en 1985 por Chaum [4] como una etiqueta privada que permite, de forma discrecional, distinguir a los participantes de una transacción. A partir de la información pública de la red, los nodos son incapaces de generar y/o vincular seudónimos para el resto de miembros de la red.

Las dificultades que atañe poner en funcionamiento un sistema de seudónimos son:

- Temporalidad: los seudónimos se tienen que renovar periódicamente porque su uso revela cierta información que se podría utilizar para identificar o localizar un sujeto.
- Generación y gestión de los seudónimos: el vínculo entre un seudónimo y la identidad real del sujeto o enlace al que está asociado es privada. Sin embargo, se deben proporcionar mecanismos para hacer llegar esta información a los usuarios autorizados de forma que la comunicación entre entidades sea viable.
- Autenticación: la autenticidad de los participantes en una transacción tendría que poder ser garantizada aunque se usasen seudónimos.

Los sistemas de seudónimos más comunes son los que utilizan una tercera entidad de confianza (TTP) responsable de generar, renovar, revocar y autenticar, los seudónimos [23], [6], [15], [7], [13]. MASK [23] sólo requiere de una TTP para distribuir las claves iniciales del sistema. Sus seudónimos son generados en base a técnicas criptográficas por pares bilineares [2], y se distribuyen fuera de línea, antes de empezar la sesión ad hoc. Los seudónimos iniciales sirven para establecer la autenticación mutua entre dos nodos, momento a partir del cual generan una lista de seudónimos y claves de enlace. La debilidad de este esquema es que se consumen considerables recursos para almacenar la información de los seudónimos (cada nodo debe disponer de un buen grupo de seudónimos ya que cada uno es solo válido para la comunicación con un vecino).

El esquema RIOMO [15] propone un sistema similar a MASK pero más eficiente, ya que los nodos pueden autogenerar sus seudónimos. Inicialmente una TTP distribuye un seudónimo raíz a todos los nodos con una clave secreta asociada. A partir de estos datos cada nodo puede generar seudónimos válidos en la red. Como MASK, RIOMO no protege la identidad del destino en los procesos de descubrimiento de rutas. Por otro lado, el sistema es poco robusto a ataques de tipo Sybil, en los que un nodo adopta varias identidades con el objetivo de participar en múltiples comunicaciones paralelas por la red y obtener más información de la topología de la misma. Ni tan solo la TTP puede tracear la relación entre los seudónimos usados en la red y los identidad real de los nodos.

Para solventar este aspecto y facilitar la autogeneración de seudónimos únicos, el esquema SPS [7] adopta un esquema jerárquico que permite que los nodos generen y revoquen sus propios seudónimos, pero impide que cada nodo tenga más un seudónimo válido durante un intervalo de tiempo. El usuario posee unas claves y certificados generados por una CA externa que son almacenados en un dispositivo *tamper-proof*. A partir de estas claves, que constituyen el nivel superior del esquema jerárquico, el usuario puede generarse unos seudónimos de segundo nivel cada vez que se necesario.

Huang propone un esquema criptográfico basado en seudónimos (*Password-based encryption*, PBE) [13], que es una versión anónima de la criptografía basada en identidades (*Identity-based encryption*, IBE). De forma análoga a IBE, los seudónimos del PBE son usados como las claves públicas de los nodos. Los nodos autogeneran los seudónimos y las correspondientes claves privadas de forma totalmente autónoma, basándose en un conjunto de parámetros públicos del sistema. Para proporcionar servicios de autorización al sistema, Huang indica que los seudónimos pueden ser certificados de forma ciega por una TTP.

Otros esquemas de generación de seudónimos no consideran el aspecto de la autorización de usuarios, y por lo tanto, el sistema es mucho más simple. Este es por ejemplo el modelo usado en ANODR [14]. En lugar de trabajar con seudónimos de identidad, ANODR utiliza seudónimos de ruta que se asocian a los enlaces salto a salto de una red ad hoc. Los seudónimos se establecen en la fase de respuesta de una acción de descubrimiento de rutas. Cada nodo genera un número aleatorio que será usado para identificar el enlace entre él, y el nodo previo de la ruta (posterior en la fase de respuesta). Los seudónimos de un enlace se pueden renovar periódicamente. Por ejemplo, si emisor y receptor están sincronizados, pueden usar una función unidireccional para derivar un nuevo seudónimo cada cierto tiempo.

III-B. Funciones unidireccionales con trampa

La funciones unidireccionales con trampa (*trapdoor functions*) son funciones unidireccionales $f : X \rightarrow Y$ tales que es fácil obtener $f(x)$ para cualquier $x \in X$, y que permiten el cálculo eficiente de la inversa (encontrar $x \in X$ tal que $f(x) = y$) si y solo si se posee cierta información adicional, la trampa. En caso contrario, el cálculo del inverso es computacionalmente intratable.

Las funciones unidireccionales con trampa se utilizan para la identificación anónima de los receptores de una comunicación. El emisor envía la información de identificación de la comunicación escondida en una función trampa, de forma que sólo el receptor legítimo de la transmisión, que posee la información trampa, sea capaz de recuperarla.

La manera más simple de implementar una función unidireccional para proporcionar anonimato de recepción es a través de criptografía de clave pública, como hacen por ejemplo los protocolos SDAR [3], ARMR [11] y AnonDSR [18]. La identidad del receptor se envía cifrada con la clave pública del propio receptor de manera que sólo él pueda abrir con éxito el paquete. Sin embargo, esta solución es muy costosa ya que el descubrimiento de rutas en redes ad hoc se hace a través de mecanismos *broadcast* de inundación, y si todos los nodos que reciben un paquete tienen que hacer una operación criptográfica para descubrir si son los receptores de un paquete, la carga total del sistema es insostenible.

De forma similar a estos últimos, los protocolos ANODR [14] y ASR [24] utilizan criptografía simétrica para esconder la identidad del destino de una comunicación. En este caso se asume que origen y destino comparten una clave TESLA [16]. El origen cifra la identidad del destino y un número aleatorio con la clave simétrica que comparten. El nodo que pueda abrir este sobre y comprobar que su identificador está en él, es el legítimo receptor. Finalmente, en el mensaje de respuesta al origen, el destino envía el número aleatorio del sobre como prueba de recepción de éste.

El protocolo ANAP [6] propone la utilización de funciones trampa más ligeras, basadas en funciones de hash. El origen identifica el destino de la comunicación a través del valor hash de su seudónimo. Sin embargo, los autores no indican cómo puede el origen obtener el seudónimo del destino.

III-C. Rutas basada en la localización

Los autores de ALARM [8] y PRISM [9] utilizan un sistema basado en la localización para establecer rutas con destinos anónimos. Los esquemas son válidos para comunicaciones que se establecen en función de las coordenadas geográficas de los nodos. Es decir, los emisores escogen el receptor no por su identidad, sino por su posición. Este tipo de comunicaciones pueden ser útiles en situaciones de desastre, en redes de vehículos VANET, etc.

Tanto ALARM como PRISM utilizan una TTP externa y fuera de línea para emitir certificados y crear firmas de grupo que ofrezcan autenticidad a los elementos del protocolo sin revelar la identidad de los nodos.

El encaminamiento en R-AO2P [20] también está basado en la localización de los nodos como método para esconder las identidades reales de los usuarios. El descubrimiento de rutas a un determinado destino se hace revelando la posición de un punto de referencia situado en la línea extendida entre el emisor y el receptor. La distancia entre el punto de referencia

y el destino es un valor aleatorio a partir del cual es difícil que un adversario pueda estimar la posición real del destino.

IV. Desvinculación de mensajes

En esta sección analizaremos las técnicas utilizadas para evitar que un adversario pueda inferir los sujetos que participan en una comunicación.

- **Mezclador** (*Mix router*): Encaminador que esconde la correspondencia entre mensajes entrantes y salientes a partir de la modificación de su apariencia y del flujo de la trasmisión.
- **Red de mezcladores** (*Mix network*): Es un conjunto de mezcladores interconectados.
- **Multicast anónimo** (*Anonymous multicast*): Todo mensaje enviado a través de una MANET puede ser recibido por cualquier nodo que se encuentre en su radio de acción. Por tanto, es básico establecer un mecanismo de este tipo para montar un sistema anónimo sobre MANETs.

IV-A. Mezclador

Un mezclador es un encaminador que recibe un conjunto de mensajes de entrada y los devuelve transformados de tal manera que no pueda relacionarse la entrada con la salida. Dichas transformaciones se producen tanto a nivel de forma (a base de aplicar técnicas de encriptación y relleno de mensajes) como de secuencia (a base de mezclar el orden y aplicar distintos retrasos en la entrega de los mensajes).

El diseño original propuesto por Chaum [5] consiste en un mezclador de proceso por lotes que almacena mensajes en la memoria del mezclador hasta que se cumple una cierta condición de descarga, momento en el que se envía el lote de mensajes desordenados. La condición de descarga puede ser una condición temporal, espacial o una combinación de ambas. La descarga temporal se establece cada cierto período de tiempo (que puede ser fijo o variable) mientras que la espacial se establece cuando se llega a sobrepasar un determinado umbral de capacidad.

El diseño original del mezclador por lotes fue extendido más adelante de forma que en el momento de la descarga solo se enviaran un subconjunto de los mensajes almacenados en el encaminador y el resto se preservaran para rondas posteriores. Dicha técnica, llamada mezclador con estanque (Pool Mix), mejora el grado de anonimato en situaciones de tráfico fluctuante a base de compensar un momento de poca carga de tráfico con un mayor retraso en la entrega de los mensajes. Esta solución es ideal para aplicaciones que no tienen restricciones de entrega muy ajustadas, tales como el correo electrónico anónimo.

En contraposición al modelo de mezclador por lotes está el mezclador continuo [10], en el que los usuarios generan un retardo aleatorio por cada mensaje que incluyen en la cabecera del mensaje. El mezclador almacena el mensaje durante el tiempo especificado y entonces lo reenvía. La ventaja de este método es que los propios usuarios controlan el tiempo límite de transferencia de la información. Este modelo funciona bien en situaciones de tráfico relativamente estable y constante. Sin embargo, en caso de que se produzcan períodos de tráfico reducido, el grado de anonimato de este modelo es bajo.

Tanto el mezclador continuo como el de estanque son vulnerables a ataques $N-1$ consistentes en la alteración del flujo de $N-1$ mensajes con el objetivo de poder trazar un mensaje concreto. Para el caso del mezclador continuo el atacante debe ser capaz de bloquear la entrada de mensajes al encaminador, mientras que para el mezclador de estanque tendría que inyectar mensajes marcados que provocaran una descarga controlada del mezclador. Para mitigar el efecto de este ataque, [11] propone que cada encaminador tome la decisión acerca del siguiente nodo sobre el que continuar la ruta, pudiendo incluso llegar a decidir añadir tráfico falso sobre rutas falsas. De hecho, las técnicas de tráfico falso son una buena manera de prevenir dicho problema (ver sección V).

IV-B. Redes de mezcladores

Para incrementar el grado de anonimato de un sistema mezclador, los enrutadores mezcladores suelen combinarse formando una red de mezcladores. De esta manera, puede llegar a preservarse el anonimato de los usuarios aún y cuando algunos nodos de la red sean comprometidos.

Tenemos dos tipologías básicas: Cascadas y mezcladores de Ruta Libre. En una Cascada, la ruta o rutas que siguen los mensajes son preestablecidas. En un mezclador de Ruta Libre, el camino a seguir por cada mensaje puede seguir una ruta independiente.

Una ventaja de las Cascadas sobre los mezcladores de Ruta Libre es el hecho de que tienden a concentrar más tráfico por sus rutas, lo que aumenta el grado de anonimato en las mismas. No obstante, en una Cascada un adversario puede llegar a conocer exactamente qué mezcladores debe controlar para trazar a un usuario en particular. Por tanto, no hay una topología mejor que las otras. Los sistemas [11], [23], [22] implementan el modelo de mezcladores de Ruta Libre, mientras que [1], [20], [14], [24], [18], [21], [17], [19], [3], [8], [9], [6], [15], [12] se ajustan al modelo de Cascada.

Por otro lado, existen modelos de mezcladores combinados que tratan de obtener las ventajas de las dos opciones, como el establecimiento de múltiples Cascadas libres.

Los sistemas de Cascada establecen una única ruta anónima, normalmente la más eficiente, sobre la que pasan todos los mensajes entre fuente y destino. Si se aplican las técnicas de un mezclador, ello puede ser suficiente para garantizar el anonimato de las transmisiones. Sobretodo considerando el hecho de que cualquier mensaje emitido a través de una MANET tiene una naturaleza multicast, lo que aumenta el conjunto de anonimato de los posibles receptores del mensaje.

Sin embargo, en caso de ataque $N-1$ un adversario global podría llegar a trazar una buena parte de la ruta analizando la evolución del tráfico en la red. Para dificultar dicho seguimiento, hay sistemas que extienden la ruta más allá de su destino o que introducen rutas falsas (ver apartado IV-C). Por otro lado, el establecimiento de una única ruta debilita la seguridad del sistema resultante al hacerlo vulnerable a ataques del tipo

rushing (en los que el adversario trata de enviar mensajes de descubrimiento de la ruta antes que el nodo fuente para tratar de «apropiarse» de la ruta) y a intrusiones de un adversario sobre uno de los nodos de la ruta.

Por ello, últimamente se están incorporando más sistemas MANET que permiten el establecimiento de circuitos a través de varias rutas.

IV-C. Multicast anónimo

En una red Ad Hoc todos los mensajes pueden ser considerados de tipo broadcast, ya que éstos pueden ser interceptados por cualquier nodo dentro del área de recepción de la señal electromagnética. Por tanto, todo mensaje enviado a través de dicho tipo de redes debe ser securizado y anonimizado en la medida de lo posible.

De cara a anonimizar los mensajes, una primera medida a tomar es eliminar o distorsionar cualquier referencia identificativa a bajo nivel, es decir, modificando las direcciones MAC de los mensajes y por ejemplo, insertando una sucesión de 1's como direcciones fuente y destino (lo que en 802.11 es indicativo de dirección multicast).

En función de la intencionalidad del emisor, podemos distinguir los siguientes tipos de mensaje:

- Multicast: mensajes dirigidos a todos los nodos en el rango de alcance directo. En su mayoría se trata de mensajes utilizados para iniciar el descubrimiento y/o mantenimiento de rutas. En este caso se trata de mensajes que contienen una parte pública (inteligible para cualquier adversario) y otra privada (utilizada para incorporar parámetros privados de establecimiento de ruta con el nodo destino).
- Unicast: mensajes dirigidos a un nodo en concreto. Son los más utilizados una vez se ha establecido una ruta. Idealmente se trata de mensajes completamente incoherentes e indistinguibles (tanto en su tamaño como contenido) por cualquier nodo que no sea aquel al que el mensaje va dirigido. Por razones de eficiencia, dichos mensajes también pueden llegar a incorporar una parte pública indicando el nodo sobre el que va dirigido el mensaje. Sin embargo, dicho parámetro de direccionamiento debería ser anónimo - es decir, sólo reconocible por el destino del mensaje.

Para evitar que nodos externos a la red puedan reconocer la parte pública de los mensajes, se puede establecer una clave que permita codificar todos los mensajes de la MANET de manera global. Normalmente, por razones de eficiencia, dicha clave global será una clave simétrica. Sin embargo, se ha de tener en cuenta que dicha medida sólo será efectiva mientras que no haya ningún intruso en la red.

Para minimizar el impacto que pueda tener la presencia de intrusos se recomienda tratar de minimizar la información topológica que pueda tener cualquier nodo de la red. Por esta razón - y también por razones de eficiencia -, los algoritmos de descubrimiento reactivos (que establecen la ruta de forma dinámica) acostumbran a ser más populares que los proactivos (en los que algunos nodos de la red son periódicamente alimentados con información topológica por parte del resto de nodos, ver [9] y [8]).

En este sentido es preferible evitar sistemas que apliquen técnicas de Onion Routing (ver Chaum [5]) para el envío de los mensajes anónimos. Ello es debido a que, para la aplicación de dicha técnica, el nodo fuente requiere codificar el mensaje a enviar utilizando N claves, una para cada nodo por los que debe pasar el mensaje. Por tanto, el nodo fuente debe conocer toda la ruta por la que debe pasar el mensaje (SDAR [3],SDDR [12]).

Una alternativa a dicha técnica consiste en el establecimiento de tablas de rutas de seudónimos asociadas a claves y seudónimos destino. Cuando un mensaje llega a un nodo proveniente de un seudónimo fuente, el mensaje es recodificado y enviado al seudónimo destino que marca su tabla. El nodo fuente tiene marcado en la tabla cuál es el seudónimo del primer nodo por el que debe pasar el mensaje para llegar a su destino. Dichas tablas son generadas por cada nodo en el momento de establecer la ruta (ver ANODR [14], ARM [17], ANONDSR [18]) o bien renovadas de forma periódica (MASK [23]).

V. Indetectabilidad

Típicamente las redes anónimas pierden robustez a lo largo del tiempo debido a que un análisis exhaustivo de las trazas de la red permite obtener información de los usuarios y las relaciones que hay entre ellos. Una forma de atacar la raíz de este problema es enmascarar los mensajes entre nodos de forma que un atacante externo no pueda distinguir cuando la red está enviando datos o ruido.

Entre las técnicas más utilizadas para enmascarar los mensajes destacamos:

- **Comunicaciones a ráfagas cortas** (*burst communications*). La transmisión de mensajes muy cortos es muy difícil de detectar por los usuarios externos a la misma. Es por ello que este tipo de transmisiones se utilizan para enviar la información de control más sensible de la red.
- **Envío de mensajes ficticios** (*dummy data*). Su objetivo es conseguir un flujo constante en la red y que el tipo de tráfico (real o falso) sea indiscernible a ojos de un atacante.
- **Modulación por espectro ensanchado** (*spread spectrum*). Las transmisiones por espectro ensanchado se caracterizan porque la información es enviada a través de un ancho de banda mucho más amplio que el mínimo requerido. Las técnicas más usadas son los sistemas de secuencia directa y los sistemas de salto de frecuencia. La ventaja de estos sistemas es que la señal es muy difícil de detectar para usuarios que desconozcan la técnica y la codificación usada para la transmisión del señal.
- **Esteganografía**. Los métodos esteganográficos permiten esconder un mensaje dentro de un flujo de comunicación cualquiera de la red, de forma que solo el receptor legítimo pueda extraer la información del canal. Para el resto de usuario el mensaje es invisible.

De las técnicas para enmascarar mensajes, las más sencilla y usada es la de envío de mensajes ficticios. Los mensajes ficticios pueden ser insertados en la entrada o salida de los

mezcladores. Normalmente la inserción en la salida provee mayor anonimato y menor retraso debido a que el mezclador puede regular de forma más precisa la introducción de mensajes ficticios en la red según el estado del tráfico [14]. Sin embargo, en el caso del ataque $N-1$ la inserción en la entrada del mezclador puede ofrecer un mayor nivel de protección.

Cuando tratamos sobre redes de mezcladores, los Mensajes Falsos pueden atravesar diversos encaminadores, tal como hace el resto de mensajes. El camino a atravesar se determina de forma aleatoria y normalmente termina en el encaminador que lo generó. Ello permite llegar a detectar ataques del tipo $N-1$ y actuar en consecuencia.

En el entorno de redes Ad Hoc, en el que el uso de recursos es muy limitado, el uso de dicho tipo de mensajes debe ser realmente minimizado. Sin embargo, también tenemos la ventaja de que un solo mensaje Falso extiende el conjunto de anonimato entre los posibles receptores a todos los nodos vecinos del nodo que emite dicho mensaje.

VI. Problemas Abiertos

Los principales problemas abiertos en el área de anonimato para redes MANET son los siguientes:

- Anonimato de los sujetos: Uno de los retos principales en esta área consiste en establecer un sistema de confianza que permita la localización y autenticación mutua de nodos que toman la identidad como atributo para seleccionar con quien establecer una comunicación, sin que ello merme el anonimato de dichos nodos frente a usuarios externos. Por tanto, se requiere un protocolo de distribución de las claves iniciales del sistema que sea eficiente y seguro.
- Métodos de incentivo: Debido a la limitada capacidad de los dispositivos deben establecerse mecanismos de incentivo y reputaciones para que los usuarios de una MANET estén dispuestos a proveer sus terminales como enrutadores de mensajes de terceros. Dichos métodos están basados en el establecimiento de protocolos de confianza distribuida, cuyo desarrollo en entornos anónimos es un reto aun no resuelto.
- Desvinculación de mensajes: Área de investigación continua para conseguir protocolos que resulten en una mayor eficiencia y robustez frente a análisis exhaustivo del tráfico.
- Indetectabilidad: Área poco explotada cuya integración con el resto de técnicas a través de soluciónes *cross-layer* podría reforzar el anonimato de los sistemas resultantes.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el Ministerio de Industria, Turismo y Comercio con el proyecto AVANZA TSI-020100-2009-374 SAT2, y por el Ministerio de Ciencia e Innovación y los fondos FEDER con los proyectos TSI2007-65406-C03-03 E-AEGIS y CONSOLIDER CSD2007-00004 ARES.

Referencias

[1] I. Aad, C. Castelluccia, J. P. Hubaux, and G. F. Switzerland. Packet coding for strong anonymity in ad hoc networks. *Proc.of IEEE SecureComm*, 6, 2006.

[2] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

[3] Azzedine Boukerche, Khalil El-Khatib, Li Xu, and Larry Korba. An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks. *Computer Communications*, 28(10):1193–1203, 2005.

[4] David Chaum. Security without identification: transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985.

[5] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, 1981.

[6] T. Ciszkowski and Z. Kotulski. Anap: Anonymous authentication protocol in mobile ad hoc networks. *Arxiv preprint cs/0609016*, 2006.

[7] Xiaojun Dang and Yang Zhang. Hierarchical pseudonym-based signature scheme and self-generated pseudonym system in ad hoc networks. *Wireless and Mobile Communications, International Conference on*, 0:282–287, 2008.

[8] K. El Defrawy and G. Tsudik. Alarm: Anonymous location-aided routing in suspicious manets. In *IEEE ICNP*. Citeseer, 2007.

[9] K. El Defrawy and G. Tsudik. Prism: Privacy-friendly routing in suspicious manets (and vanets). In *IEEE International Conference on Network Protocols, 2008. ICNP 2008*, pages 258–267, 2008.

[10] C. Diaz and B. Preneel. Taxonomy of mixes and dummy traffic. In *Information security management, education and privacy: IFIP 18th World Computer Congress: TC11 19th International Information Security Workshops, 22-27 August 2004, Toulouse, France*, pages 217–223. Kluwer Academic Pub, 2004.

[11] Ying Dong, Tat Wing Chim, Victor O. K. Li, S. M. Yiu, and C. K. Hui. Armr: Anonymous routing protocol with multiple routes for communications in mobile ad hoc networks. *Ad Hoc Networks*, 7(8):1536–1550, 11 2009.

[12] K. El-Khatib, L. Korba, R. Song, and G. Yee. Secure dynamic distributed routing algorithm for ad hoc wireless networks. In *Parallel Processing Workshops*, pages 359–366, 2003.

[13] Dijiang Huang. Pseudonym-based cryptography for anonymous communications in mobile ad hoc networks. *Int. J. Secur. Netw.*, 2(3/4):272–283, 2007.

[14] J. Kong and X. Hong. Anodr: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 291–302. ACM New York, NY, USA, 2003.

[15] S. Md, M. Rahman, A. Inomata, T. Okamoto, and M. Mambo. Anonymous secure communication in wireless mobile ad-hoc networks. In *Proceedings of the First International Conference on Ubiquitous Convergence Technology*, pages 131–140. Citeseer, 2006.

[16] Adrian Perrig, Ran Canetti, J.D. Tygar, and Dawn Song. The tesla broadcast authentication protocol. *RSA CryptoBytes*, 5(2):2–13, 2002.

[17] S. Seys and B. Preneel. Arm: Anonymous routing protocol for mobile ad hoc networks. *International Journal of Wireless and Mobile Computing*, 3(3):145–155, 2009.

[18] R. Song, L. Korba, and G. Yee. Anondsr: efficient anonymous dynamic source routing for mobile ad-hoc networks. In *Proceedings of the 3rd ACM Workshop on Security of Ad hoc and Sensor Networks*, page 42. ACM, 2005.

[19] D. Sy, R. Chen, and L. Bao. Odar: On-demand anonymous routing in ad hoc networks. In *Proc. of IEEE MASS*, pages 267–276. Citeseer, 2006.

[20] X. Wu and B. Bhargava. Ao2p: Ad hoc on-demand position-based private routing protocol. *IEEE Transactions on Mobile Computing*, pages 335–348, 2005.

[21] L. Yang, M. Jakobsson, and S. Wetzel. Discount anonymous on demand routing for mobile ad hoc networks. In *Proc. of the Second International Conference on Security and Privacy in Communication Networks (SECURECOMM)*. Citeseer, 2006.

[22] Y. Zhang, W. Liu, and W. Lou. Anonymous communications in mobile ad hoc networks. In *IEEE INFOCOM*, volume 3, pages 1940–1951. Citeseer, 2005.

[23] Y. Zhang, W. Liu, W. Lou, and Y. Fang. Mask: Anonymous on-demand routing in mobile ad hoc networks. *IEEE transactions on wireless communications*, 5(9):2376, 2006.

[24] B. Zhu, Z. Wan, M. S. Kankanhalli, F. Bao, and R. H. Deng. Anonymous secure routing in mobile ad-hoc networks. In *29th IEEE International Conference on Local Computer Networks (LCN)*, pages 102–108. Citeseer, 2004.

Ofuscación del perfil del usuario de un motor de búsqueda mediante una red social y protocolos criptográficos

Arnau Erola Cañellas, Josep Domingo-Ferrer and Jordi Castellà-Roca
Universitat Rovira i Virgili,
Dpt. d'Enginyeria Informàtica i Matemàtiques,
Cátedra UNESCO de Privacidad de Datos,
Av. Països Catalans 26,
E-43007 Tarragona
Email: {arnau.erola, josep.domingo, jordi.castella}@urv.cat

***Resumen*—Los motores de búsqueda de Internet crean perfiles de sus usuarios con la finalidad de ofrecerles un servicio más personalizado. En el año 2006, el escándalo AOL, motivado por la publicación de las consultas de 658.000 usuarios, puso en duda la correcta protección de esta información y causó preocupación a los usuarios. De ahí que hayan aparecido numerosas propuestas para proteger la privacidad de los usuarios, la más importante de las cuales es la recuperación de información con privacidad (en inglés, *Private Information Retrieval* o PIR). Sin embargo, las dificultades de implementación que presenta el PIR estricto hacen necesaria su relajación en la práctica. Esta relajación del PIR se conoce como recuperación de información con privacidad de usuario (en inglés *User-Private Information Retrieval* - UPIR). En la mayoría de protocolos UPIR, varios usuarios colaboran entre ellos para proteger su privacidad. Las últimas propuestas implementan los sistemas UPIR sobre redes sociales, aprovechando su estructura y evitando el coste de mantenimiento de una red P2P. Aún así, la sobrecarga de la red o la pérdida de privacidad frente a otros usuarios sigue siendo común. A continuación presentamos tres protocolos de UPIR basados en redes sociales y distintas técnicas criptográficas.**

***Index Terms*—Recuperación de información con privacidad (*Private information retrieval*), Recuperación de información con privacidad de usuario (*User-private information retrieval*), Redes sociales (*Social networks*), Protocolos criptográficos (*Cryptographic protocols*).**

I. Introducción

El objetivo de la recuperación de información con privacidad (en inglés *Private Information Retrieval* - PIR) es permitir a un usuario recuperar un objeto de una base de datos sin que posteriormente se pueda conocer de qué objeto se trataba. En la literatura de PIR (ver los documentos seminales [1], [2] y el resumen [3]) la base de datos se modela como un vector, y se supone que el usuario quiere recuperar el i-ésimo componente del vector, manteniendo el índice i oculto ante la base de datos. Aunque esta visión de los protocolos PIR permite desarrollos teóricos, se basa en muchas suposiciones que dificultan su desarrollo práctico:

- En general, la base de datos no puede modelarse como un vector en el cual el usuario conoce la ubicación física i del objeto que le interesa (por ejemplo, pensemos en un usuario consultando un motor de búsqueda).
- Si la base de datos contiene n objetos, los protocolos PIR teóricos tienen una complejidad $O(n)$ [1], [2]: el protocolo debe "acceder" a todos los registros para evitar dar alguna pista al servidor sobre el valor de i; esto es inasequible para grandes bases de datos o motores de búsqueda [4];
- Se supone que el servidor de la base de datos coopera en el protocolo PIR; no obstante, es el usuario quien está interesado en su propia privacidad, mientras que la motivación del servidor de la base de datos es dudosa; de hecho, PIR parece poco atractivo para la mayoría de compañías que ofrecen bases de datos consultables, ya que limita su habilidad para crear perfiles de usuario.

Por las razones anteriores, parece ser necesario relajar las suposiciones del PIR por el bien de su aplicación práctica. A continuación repasamos algunas de las relajaciones propuestas.

En primer lugar, notemos que los sistemas de encaminamiento por capas (*onion-routing*) como Tor [5] no tienen por objetivo la recuperación de información con privacidad. Estos sistemas protegen el transporte de los datos, pero no ofrecen una protección punto a punto a nivel de aplicación. Siempre que un motor de búsqueda o un servidor de base de datos pueda enlazar las consultas sucesivas sometidas por el mismo usuario (*p.e* usando galletas), podrá perfilar e identificar al usuario. Del mismo modo, los sistemas basados en hardware de confianza como [6], solo trasladan el problema de la base de datos a dicho hardware.

En [7] se propone un sistema llamado GooPIR, en el cual un usuario enmascara su consulta real añadiéndole $k-1$ consultas falsas y luego sometiendo la consulta enmascarada resultante a un motor de búsqueda o una base de datos grande que no tiene por qué cooperar (de hecho, no tiene ni por qué saber que el usuario está intentando proteger su privacidad). *Strictu sensu*, GooPIR no proporciona PIR como se ha definido anteriormente; más bien proporciona recuperación de información con $h(k)$-privacidad, en la que se esconde

la consulta real dentro de un conjunto de k consultas de entropía al menos $h(k)$. Este sistema funciona adecuadamente pero supone que las frecuencias de las palabras y frases que aparecen en una consulta son conocidas y están disponibles: para mayor privacidad, las frecuencias de las consultas reales y falsas deben ser similares, de modo que la incertidumbre $h(k)$ del motor de búsqueda sobre la consulta real sea máxima.

TrackMeNot [8] es otro sistema práctico basado en un principio diferente: en lugar de someter una única consulta enmascarada para cada consulta real, como hace GooPIR, esconde la consulta real del usuario en una nube de consultas-fantasma sometidas de forma automática en diferentes intervalos de tiempo a varios motores de búsqueda. Su funcionamiento es transparente para el usuario, dado que es una extensión del navegador y realiza las consultas cuando la actividad del usuario es baja. Aunque este sistema resulta práctico a pequeña escala, si su uso se generalizase, la sobrecarga introducida por las consultas-fantasma degradaría significativamente el rendimiento de los motores de búsqueda y las redes de comunicación. Además, los intervalos entre sumisiones de consultas-fantasma pueden ser diferentes de los intervalos entre consultas reales, lo que puede dar pistas al motor de búsqueda para identificar este último tipo de consultas.

Crowds [9] propone proteger la privacidad de los usuarios de una red ocultando sus acciones mediante las de otros usuarios. Un usuario puede someter una consulta al motor de búsqueda o mandarla a otro usuario de la red. A su vez, éste puede someter la consulta o mandarla a otro usuario; y así sucesivamente, hasta que un usuario someta la consulta al motor de búsqueda. Una debilidad del sistema es que la entrada y salida de usuarios requiere un nodo central. Además, el reenvío de mensajes utiliza una probabilidad fija que permite identificar al usuario origen, ya que sus mensajes no se distribuyen uniformemente por la red. Tampoco hay ningún control sobre el envío de mensajes, siendo el sistema susceptible a ataques de denegación de servicio.

Con la misma finalidad práctica, el sistema descrito en [10] oculta a los usuarios en una comunidad anónima de usuarios por pares (P2P). Un usuario somete consultas en nombre de sus vecinos anónimos y viceversa. Los usuarios en la comunidad P2P comparten claves simétricas de cifrado que utilizan para establecer canales confidenciales. De este modo, la base de datos conoce la consulta que se está recuperando (lo que se diferencia del PIR estricto), pero no puede obtener los historiales de búsqueda de los usuarios, ya que quedan difuminados entre los usuarios vecinos. Este enfoque tiene algunas ventajas: a diferencia de [7], no requiere conocer la frecuencia de todas las posibles palabras y frases que pueden ser consultadas, y, a diferencia de [8], no sobrecarga al motor de búsqueda con consultas-fantasma. Sobre la preservación de la privacidad del perfil del usuario respecto a la base de datos y a intrusos externos, el sistema [10] ofrece privacidad hacia los usuarios vecinos porque los vecinos son anónimos entre sí. Algunos inconvenientes del sistema son la necesidad de crear una red P2P anónima y la distribución de claves en dicha red, aunque en [10] se muestra como distribuir claves sin una tercera parte de confianza.

En [11], se propone otro enfoque UPIR P2P. En este caso, el anonimato se consigue con la ayuda de un nodo central que pone en contacto a un grupo de n usuarios que quieren someter una consulta cada uno. Los usuarios ejecutan un protocolo de recuperación de consultas donde cada usuario recibe una consulta de uno de los restantes $n-1$ usuarios, sin saber de qué usuario proviene. Cada usuario somete la consulta recibida al motor de búsqueda, y manda por difusión a todos los usuarios la respuesta obtenida. De este modo, los usuarios que publicaron la consulta pueden recuperar la correspondiente respuesta. Aunque este esquema preserva la privacidad del historial de consultas frente a los vecinos, tiene el defecto de que todas las respuestas son vistas por todos los vecinos, lo que es una desventaja respecto a los esquemas anteriormente citados.

En [12] se presenta una alternativa al P2P UPIR basada en una red social. Un usuario que quiere hacer una consulta la envía a uno de sus vecinos de la red social. Este vecino puede someter la consulta directamente a la base de datos o mandarla a uno de sus vecinos; y así sucesivamente, hasta que uno de ellos someta la consulta. La respuesta a la consulta sigue el camino inverso hacia el usuario que la ha originado. Para proteger el anonimato del origen de la consulta, la propuesta distribuye las consultas uniformemente entre los usuarios. Una característica atractiva de este sistema es el uso de una red social existente como comunidad de pares, lo que facilita su implementación. Sin embargo, un problema de esta propuesta es que la consulta sometida es vista por todos los usuarios que la han aceptado, es decir, desde usuario origen hasta el usuario que la envía al motor de búsqueda. Algunos usuarios pueden preferir someter ciertas consultas directamente a un motor de búsqueda antes que revelárselas a sus amigos.

I-A. Contribución y plan de este artículo

Presentamos protocolos criptográficos para recuperación de información con privacidad de usuario que reúnen las ventajas de [10] y [12], a la vez que mejoran sus puntos débiles.

En la sección II se presentan las hipótesis y el objetivo de nuestro sistema. En la sección III se describe el funcionamiento de nuestras propuestas y a su vez se comentan sus ventajas e inconvenientes. La sección IV contiene algunas conclusiones.

II. Hipótesis y objetivo

Las redes sociales son un fenómeno en crecimiento [13]. Una red social es una comunidad de usuarios con intereses en común donde cada uno puede publicar o compartir información y servicios. Concretamente, seguiremos el concepto de red social descrito en [14]. Esta propuesta no requiere la existencia de un nodo central. Así, los usuarios están conectados directamente entre ellos o por medio de otros usuarios que actúan como intermediarios. Cada usuario puede estar en línea o desconectado en un momento dado. Por ese motivo, debemos suponer que se trata de una red *ad hoc*.

Consideraremos además que una red social es una red abierta a la exploración. Esto significa que los usuarios conocen la topología de la red social (o gran parte de ella).

Por consiguiente, los usuarios son capaces de calcular las distancias entre ellos (número de saltos entre cada par de usuarios).

Sea U_i un usuario perteneciente a una red social RS y $\{N_1, \ldots, N_k\}$ su conjunto de vecinos (un vecino es una relación directa en la red social). U_i quiere someter una consulta a un motor de búsqueda sin que éste cree un perfil detallado suyo, o lo que es lo mismo, U_i quiere mantener su privacidad. Con esta finalidad, U_i obtiene la ayuda de los demás usuarios de la red social para que sometan su consulta en su nombre.

El objetivo de nuestra propuesta es proteger la privacidad de los usuarios conectados a una red *ad hoc* frente a un motor de búsqueda y también frente a los otros usuarios de la red social. Para conseguirlo hacemos uso de la criptografía.

III. Protocolos criptográficos para ofuscación del perfil de usuario

A continuación vamos a presentar tres esquemas distintos:

- Clave pública.
- Clave simétrica.
- Esquema de umbral.

En todos los esquemas vamos a considerar a un usuario U_i, perteneciente a una red social RS, que quiere someter una consulta q a un motor de búsqueda. Los usuarios de RS conocen la topología de la red pero solamente se pueden comunicar con sus relaciones directas. Nótese que el funcionamiento y mantenimiento de la red social están fuera de los objetivos de este trabajo.

III-A. Clave pública

En esta propuesta cada usuario de la red social dispone de un par de claves (clave pública PK - clave privada SK) y del correspondiente certificado digital.

Los usuarios que quieren entrar en el sistema mandan peticiones a sus vecinos pidiéndoles su certificado digital. Esta petición se propaga por la red hacia el resto de usuarios hasta un máximo de saltos. La primera vez que el usuario utiliza el sistema debe esperar a recibir su primer certificado para poder participar.

Cuando un usuario quiere someter una consulta, la manda junto con una clave simétrica, obtenida aleatoriamente, por el camino más corto hacia otro usuario de quien desea ayuda. Esta información está cifrada con la clave pública del destinatario, de modo que sólo éste puede conocer el contenido del mensaje. La respuesta a la consulta llega al usuario origen cifrada con la clave simétrica adjunta a la consulta.

III-A1. Acceso al sistema: Sea c_r un mensaje definido e identificable de nuestro sistema que se utiliza como petición de certificados y sea c_a su respuesta. Cada usuario mantiene una tabla de certificados recibidos de otros usuarios, de ahora en adelante denominada τ. Cada fila de τ representa a un usuario y contiene su identificador id (el identificador de su certificado digital), su clave pública PK_{id}, su certificado digital $CERT_{id}$ y el contador de disponibilidad $count$. Para evitar la saturación de la red, c_r contiene un contador de saltos, TTL. En cada salto se decrementa el TTL y, mientras sea mayor que cero, c_r es reenviado.

Si el usuario U_i quiere entrar en el sistema, ejecutará el siguiente protocolo:

- Cada cierto intervalo de tiempo w, U_i manda a todos sus vecinos una petición c_r.
- Cuando U_i recibe una c_a:
  - Comprueba la validez del certificado recibido.
    - Si es válido lo agrega a τ. Si ya existe el usuario en τ, incrementa su contador de disponibilidad $count$.
    - Si el certificado no es válido, lo ignora.
- Al finalizar el periodo w, todos los contadores de disponibilidad de los usuarios son decrementados con el fin de mantener actualizada la tabla con los usuarios que están en línea. Cuando el contador de disponibilidad de un usuario llega a 0, el usuario es eliminado de la tabla. Si todos los usuarios son borrados de τ, U_i queda aislado.

Cuando un usuario recibe una petición c_r realiza los pasos siguientes:

- Responde con su certificado al vecino que le ha enviado c_r.
- Decrementa el TTL de c_r.
- Si el $TTL > 0$ reenvia c_r a sus vecinos (excepto al vecino que le ha enviado c_r).

III-A2. Sumisión de una consulta: Cuando el usuario U_i, con identificador id_i, quiere someter la consulta q a un motor de búsqueda, ejecuta el protocolo siguiente:

- Escoge aleatoriamente a un usuario de su tabla τ. Sea U_f el usuario seleccionado por U_i e id_f su identificador.
- Obtiene una clave simétrica sk de forma aleatoria.
- Crea el mensaje $M = q_{id}, E_{PK_f}(q, sk), id_f$ donde q_{id} es el identificador de la consulta; $E_{PK_f}(q, sk)$ son la consulta q y la clave simétrica sk cifrados bajo la clave pública PK_f del destinatario; id_f es el identificador del destinatario.
- Inicia el temporizador t.
- Manda el mensaje M al vecino más cercano a U_f, por ejemplo U_j. U_j manda el mensaje a su vecino más cercano a U_f, y así sucesivamente hasta que el mensaje llega a U_f. De este modo, M llaga a U_f por el camino más corto.
- Espera un tiempo máximo t hasta recibir respuesta del usuario U_f:
  - Así que la respuesta llega, finaliza el protocolo.
  - Si la respuesta no llega en este tiempo, vuelve a ejecutar el protocolo.

Cuando un usuario recibe una consulta de otro usuario, ejecuta los siguientes pasos:

- Si es el destinatario del mensaje recibido:
  - Descifra el mensaje $M = q_{id}, D_{SK_f}(q, sk), id_f$
  - Envía la consulta q al motor de búsqueda.
  - Cifra la respuesta del motor de búsqueda con sk.

- Manda la respuesta al usuario origen siguiendo el camino inverso.

- Si la consulta va destinada a otro usuario U_f, la reenvía a su vecino más cercano a U_f.

III-A3. Análisis del protocolo: Este esquema permite conocer el número de saltos que hace una consulta hasta que llega a su destinatario, con lo que se puede determinar el tiempo estimado que tardará [15]. Si se quiere reducir este tiempo, el usuario puede mandar la consulta a un usuario más cercano, aumentando la usabilidad del sistema, pero reduciendo la privacidad, pues el tamaño del grupo será inferior.

Además, el esquema no requiere que los usuarios acuerden previamente una clave común. Sin embargo, el uso de claves públicas obliga a cada usuario a mantener un gran número de certificados digitales, concretamente uno por cada usuario conocido en la red.

Como se ha comentado en la sección I, tan importante como la privacidad frente al motor de búsqueda es la privacidad frente a otros usuarios de la red. Al encaminar tanto la consulta como su respuesta salto a salto, el origen de la consulta se mantiene oculto ante los otros usuarios. Sin embargo, sí se conoce la identidad (id_f) del destinatario.

III-B. Clave simétrica

Nuestra segunda propuesta pretende reducir el número de certificados que cada usuario tiene que almacenar, y a la vez proteger el anonimato de los usuarios mediante la formación de grupos. Los usuarios de cada grupo escogen una clave en común g_k de forma distribuida (sin autoridad central) siguiendo el esquema propuesto en [10], que usa un diseño combinatorio llamado configuración para incrementar la disponibilidad en el sistema y reducir el número de claves requeridas. Las configuraciones permiten distribuir un número v de claves entre b usuarios, con $1 \leq v \leq b(b-1)/2$, de tal modo que cada usuario recibe el mismo número de claves y cada clave es compartida por el mismo número de usuarios.

III-B1. Acceso al sistema: Cuando el usuario U_i quiere formar un grupo ejecuta las siguientes acciones:

- Publica que quiere formar un grupo de g usuarios.
- Acepta peticiones hasta que recibe $g-1$. Por cada petición aceptada, U_i manda un mensaje de confirmación.

Los vecinos de U_i interesados en formar parte del grupo le mandan un mensaje g_r. Si un vecino U_j es aceptado en el grupo, recibe un mensaje de confirmación y posteriormente publica que forma parte de un grupo con U_i. Desde ese momento, los vecinos de U_j pueden intentar unirse al grupo mandando una petición g_r a U_i por el camino inverso, esto es, pasando por U_j.

Una vez el grupo está completo, los usuarios acuerdan una clave simétrica y privada común (g_k) con el algoritmo distribuido [10]. Esta clave permitirá a los usuarios del grupo comunicarse de forma segura entre ellos.

Notese que los vecinos de U_i en el grupo son sus vecinos en RS que también pertenecen al grupo.

III-B2. Sumisión de una consulta: Cada usuario puede formar parte de varios grupos. Para someter una consulta, escoge uno de estos grupos y repite el protocolo de la sección III-A2, sustituyendo PK_f por g_k y enviando el mensaje a todos sus vecinos de RS pertenecientes al grupo, en lugar de a un usuario concreto. El mensaje enviado contiene la misma información que en el protocolo de la sección III-A2 menos el identificador del destinatario, pues en este caso no es un usuario concreto. Esto es, $M = q_{id}, E_{g_k}(q, sk)$.

Cada usuario que recibe una consulta realiza los pasos siguientes:

- Reenvía la consulta a sus vecinos del grupo.
- Decide aleatoriamente si responde a la consulta o no.
  - Si decide responder la consulta, ejecuta los pasos siguientes:
    - Descifra el mensaje $M = q_{id}, D_{g_k}(q, sk)$.
    - Somete la consulta q al motor de búsqueda.
    - Cifra la respuesta del motor de búsqueda con sk.
    - Envía la respuesta por difusión a sus vecinos del grupo.
  - Si decide no responder la consulta, ignora la petición.

III-B3. Análisis del protocolo: En este esquema un usuario puede formar parte de varios grupos. Por ello, el usuario tiene que almacenar las claves simétricas de cada grupo al que pertenece. El hecho de guardar solamente una clave simétrica por grupo, en lugar de una clave pública y un certificado por cada usuario conocido, reduce el almacenamiento requerido con respecto al esquema de clave pública.

Sin embargo, el uso del grupo también supone una pérdida de privacidad, puesto que todos los usuarios del grupo conocen la consulta sometida, aunque no sea identificable su origen (los emisores de las llamadas "vanity queries", en las cuales un usuario se busca a sí mismo en el web, son obviamente identificables).

El envío de mensajes por difusión permite que los usuarios permanezcan anónimos. Sin embargo, aunque a pequeña escala pueda funcionar bien, a medida que el número de usuarios crece, se degrada el rendimiento de la red de comunicación. Análogamente, el aumento de usuarios supone un mayor número de consultas mandadas al motor de búsqueda (nótese que cada consulta puede ser sometida por varios usuarios del mismo grupo), ocasionando una posible sobrecarga.

La recepción de varias respuestas a su consulta permite al usuario comprobar la veracidad de las mismas. Si un usuario devuelve una respuesta falsa a otro usuario, la respuesta falsa podrá ser detectada si el resto de respuestas son correctas.

La formación del grupo es un proceso que puede resultar costoso. Primero se deben reunir suficientes usuarios y posteriormente acordar una clave común. Este proceso sería poco usable si se tuviera que formar un grupo por cada consulta a someter. En cambio, como el grupo puede permanecer formado durante mucho tiempo, el coste de su formación puede considerarse aceptable.

III-C. Sistema de umbral

Esta última propuesta pretende solucionar los problemas de degradación de la red de comunicación y la sobrecarga del motor de búsqueda que presenta el esquema de clave simétrica mediante un criptosistema de umbral (t, l)-ElGamal [16]. En un esquema de umbral (t, l), un conjunto de l usuarios acuerda una clave pública común PK y cada usuario recibe una parte de la clave secreta SK_i, de tal modo que cualquier subconjunto de t usuarios puede descifrar un mensaje cifrado con PK, pero menos de t usuarios no pueden.

Cuando un usuario quiere someter una consulta q, escoge uno de los grupos a los que pertenece. A continuación cifra la consulta q y la clave secreta sk con la clave pública PK del grupo, y envía el criptograma obtenido a un vecino en RS perteneciente a ese grupo. El vecino lo descifra y, si no puede leer el mensaje, lo manda a otro vecino. Y así sucesivamente durante t saltos. La respuesta a la consulta vuelve por el camino inverso hacia el usuario que ha originado q.

III-C1. Acceso al sistema: Cuando un usuario quiere entrar en el sistema ejecuta el protocolo de la sección III-B1, con la única diferencia de que utiliza el sistema presentado en [16] para obtener la clave pública de cifrado y las partes de la clave privada de descifrado.

Igual que en el esquema anterior, cada usuario puede formar parte de varios grupos y sus vecinos en cada grupo son sus vecinos en RS que también pertenecen a ese grupo. Además, cada usuario tiene que almacenar la clave pública y su parte de la clave privada de cada grupo al que pertenece.

III-C2. Sumisión de una consulta: Si el usuario U_i quiere enviar una consulta q al motor de búsqueda ejecuta los pasos siguientes:

- Escoge un grupo G y un vecino $U_j \in G$
- Obtiene una clave simétrica sk.
- Manda a U_j el mensaje $M = q_{id}, G, E_{PK_G}(q, sk)$, donde PK_G es la clave pública del grupo G.
- Inicia un temporizador y espera la recepción de la respuesta. Si el temporizador expira sin recibir la respuesta repite el proceso con otro grupo y vecino.

Un usuario al recibir una petición de consulta ejecuta los siguientes pasos:

- Descifra el mensaje recibido $m = q_{id}, G, D_{SK_{G,i}}(q, sk)$, donde $SK_{G,i}$ es la parte de la clave secreta del grupo G que posee el usuario.
- Si m es un mensaje válido:
  - Envía la consulta q al motor de búsqueda.
  - Cifra la respuesta del motor de búsqueda con sk.
  - Manda la respuesta al usuario origen por el camino inverso.
- Si m no és un mensaje válido, lo reenvía a un vecino en RS perteneciente a G.

III-C3. Análisis del protocolo: En este esquema todos los usuarios permanecen anónimos. Sin embargo, al utilizar un criptosistema de umbral (t, l)-ElGamal, sabemos que son necesarios t saltos para poder descifrar la consulta, lo que nos puede dar una idea aproximada del origen de la consulta. Podemos incrementar la incertidumbre permitiendo que el usuario origen descifre la consulta cifrada una primera vez o no. De éste modo no se podrá saber con certeza si el mensaje ha hecho t o $t-1$ saltos.

Gracias a que se ha evitado el envío de mensajes por difusión, nuestro sistema no sufre de saturación. A su vez, esto nos permite saber el tiempo de respuesta estimado, ya que se conoce el número de saltos que hace un mensaje [15].

Un nuevo aspecto interesante es la imputación de revelación de clave. En el esquema de clave simétrica, los usuarios compartían una misma clave. En caso de que ésta fuera revelada no se podía saber quién había sido. Este último esquema permite saber qué usuarios han revelado su clave secreta, pues cada uno tiene una distinta.

IV. Conclusiones

Los tres protocolos presentados (clave pública, clave simétrica, y sistema de umbral) garantizan la protección del perfil de los usuarios frente al motor de búsqueda y frente al resto de usuarios de la red social. El protocolo basado en criptografía de clave pública permite conocer el número de saltos que realizará la consulta, de manera que se puede estimar el tiempo de respuesta. El protocolo de clave simétrica es adecuado en un entorno con atacantes que manipulan las respuestas del motor de búsqueda. En este caso, el usuario que ha originado la consulta recibe varias respuestas, y puede detectar una manipulación si las respuestas no coinciden. Finalmente, el protocolo basado en un esquema de umbral reduce el número de claves públicas que se deben almacenar y no sobrecarga al motor de búsqueda.

Los protocolos presentados pueden ser implementados en un mismo sistema como extensión de un navegador web. El usuario, según sus necesidades, escogerá utilizar uno u otro.

Agradecimientos

Los autores agradecen las ayudas del MICINN (proyectos "E-AEGIS" TSI2007-65406-C03-01, "ARES" CONSOLIDER INGENIO 2010 CSD2007-0004), del Ministerio de Industria, Comercio y Turismo (proyecto TSI-020100-2009-720), y de la Generalitat de Catalunya (ayuda 2009 SGR 1135). Josep Domingo-Ferrer recibe financiación de la Generalitat de Catalunya como Premiado ICREA Academia. Los autores se encuadran en la Cátedra UNESCO de Privacidad de Datos, pero sus ideas no reflejan necesariamente la posición de la UNESCO ni comprometen a dicha organización.

Referencias

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *FOCS '95: Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, 1995.

[2] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *Journal of the ACM*, volume 45, pages 965–981, 1998.

[3] Rafail Ostrovsky and William Skeith. A survey of single-database private information retrieval: Techniques and applications. In *Public Key Cryptography - PKC 2007*, pages 393–411, 2007.

[4] Amos Beimel, Yuval Ishai, and Tal Malkin. Reducing the servers computation in private information retrieval: PIR with preprocessing. *J. Cryptol.*, 17(2):125–151, 2004.

[5] Tor project, 2009. `http://www.torproject.org`.

[6] Yanjiang Yang, Xuhua Ding, Robert H. Deng, and Feng Bao. An efficient pir construction using trusted hardware. In *ISC '08: Proceedings of the 11th international conference on Information Security*, pages 64–79. Springer-Verlag, 2008.

[7] J. Domingo-Ferrer, A. Solanas, and J. Castellà-Roca. h(k)-private information retrieval from privacy-uncooperative queryable databases. In *Online Information Review*, volume 33, pages 720–744, 2009.

[8] D.C. Howe and H. Nissenbaum. Trackmenot: resisting surveillance in web search. Lessons from the Identity Trail: Privacy, Anonymity and Identity in a Networked Society, pages 409–428. Oxford University Press, 2009.

[9] A.D. Rubin M.K. Reiter. Crowds: anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1):66–92, 1998.

[10] J. Domingo-Ferrer, M. Bras-Amorós, Qianhong Wu, and J. Manjón. User-private information retrieval based on a peer-to-peer community. In *Data and Knowledge Engineering*, volume 68(11), pages 1237–1252, 2009.

[11] Jordi Castellà-Roca, Alexandre Viejo, and Jordi Herrera-Joancomartí. Preserving user's privacy in web search engines. *Comput. Commun.*, 32(13-14):1541–1551, 2009.

[12] A. Viejo and J. Castellà-Roca. Using social networks to distort users' profiles generated by web search engines. In *Computer Networks*, volume 54, pages 1343–1357, 2010.

[13] Social networking boom as good as over by 2009, 2007. `http://www.daniweb.com/blogs/entry1753.html`.

[14] J. Domingo-Ferrer. A public-key protocol for social networks with private relationships. *LNCS*, 4617:373–379, 2007.

[15] Arnau Erola, Jordi Castellà-Roca, and Alex Viejo. Exploiting social networks for improving privacy in personalized web search. 2010. Manuscript.

[16] P. Fouque and J. Stern. One round threshold discrete-log key generation without private channels. In *PKC '01: Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography*, pages 300–316, 2001.

Eficiencia y Privacidad en una Mixnet Universalmente Verificable

Jordi Puiggalí Allepuz
Scytl Secure Electronic Voting
Email: jordi.puiggali@scytl.com

Sandra Guasch Castelló
Scytl Secure Electronic Voting
Email: sandra.guasch@scytl.com

***Resumen*—Los procesos de auditoría transparentes son importantes en las elecciones electrónicas. Las mixnets universalmente verificables ayudan a conseguir esta transparencia generando pruebas criptográficas que pueden ser verificadas por un auditor. En este artículo presentamos un sistema eficiente de verificación de mixnets que consigue un buen nivel de precisión a la vez que preserva totalmente la privacidad de los votantes. Además, se propone un método para cifrar los votos que mejora la capacidad de detección de errores en el cifrado a la vez que se mantiene el rendimiento de la mixnet.**

I. Introducción

Cuando una elección se lleva a cabo de forma electrónica, el principal problema que surge es cómo implementar un proceso de auditoría transparente. En las elecciones tradicionales, existen auditores independientes y observadores que pueden supervisar directamente el proceso de la elección en tiempo real, especialmente el proceso de apertura de las urnas y el recuento de los votos. Cuando el recuento se realiza de forma electrónica (por ejemplo, el descifrado y recuento de los votos), la supervisión del proceso lógico durante su ejecución en la máquina es prácticamente imposible, ya que no puede ser monitorizado directamente por el humano como en las elecciones tradicionales. Así pues, es muy importante que un sistema de voto electrónico proporcione métodos de auditoría transparentes para poder verificar su correcto funcionamiento.

Dado que en el voto electrónico los resultados se pueden verificar haciendo un recuento paralelo de los votos (de la misma forma que cuando la elección se realiza de forma tradicional), la dificultad del proceso de auditoría reside en la verificación de la correcta apertura de los votos, es decir, el proceso de descifrado.

Una posible solución consiste en permitir que los auditores u observadores instalen programas en el sistema para monitorizar la plataforma de voto. El problema reside en que estos programas de monitorización deberían ser también monitorizados a su vez, dado que el proceso de descifrado podría ser vulnerable a ellos. Como podemos ver, este procedimiento introduce un bucle infinito que no tiene fácil solución (*who watches the watchmen?*).

Otra solución alternativa podría ser auditar el proceso de descifrado monitorizando la información de logs generada durante su ejecución. No obstante, si asumimos que el proceso de descifrado puede ser comprometido, la información de logs podría ser manipulada a su vez para ocultar prácticas maliciosas. Es más, la información de logs que proporciona el proceso de descifrado debe ser limitada para preservar la privacidad de los votantes (por ejemplo, no se debería registrar la relación entre un contenido descifrado y un voto cifrado si el último está conectado a la identidad de un votante).

En 1995, Sako y Kilian [SK95] introdujeron el concepto de *verificabilidad universal* en su propuesta de un proceso de descifrado de votos basado en una mixnet. Esta propiedad consiste en proporcionar medios a cualquier auditor u observador para verificar el correcto descifrado de los votos, utilizando pruebas criptográficas que son generadas por el proceso de descifrado.

Una mixnet está compuesta por uno o varios nodos que mezclan los mensajes de entrada utilizando una permutación cuyo valor es guardado en secreto. Los nodos de la mixnet también realizan un proceso de transformación que modifica los valores de los cifrados de los votos en la entrada para evitar la correlación de las entradas y salidas (protegiendo así la privacidad de los votantes). Dado que esta transformación no permite asegurar a simple vista que los votos en la salida del nodo de la mixnet son los mismos que los votos en la entrada, es importante poder verificar el proceso de mixing y descifrado de un modo que preserve la privacidad de los votantes y la integridad de los votos.

Desde que Chaum presentó la primera mixnet en 1981 [Ch81], la búsqueda de métodos eficientes de verificación que no pongan en peligro el proceso de anonimización de los votos en el mixing (por ejemplo revelando la permutación secreta o los valores de recifrado) es una área muy fértil de investigación. Concretamente, la propiedad de verificabilidad universal es el propósito principal de las mixnets diseñadas en los últimos 15 años.

En este artículo presentamos un método de verificación universal eficiente para mixnets de recifrado que tiene una gran capacidad de detección de modificaciones a la vez que se preserva la privacidad de los votantes. El artículo se estructura de la siguiente forma: en la sección II explicamos nuestra motivación para diseñar un nuevo sistema de verificación de mixnets, en la sección III se define el criptosistema utilizado para cifrar los votos, en la sección IV se presenta el nuevo método de verificación de mixnets, en la sección V se propone un sistema para cifrar los votos de forma eficiente, y las conclusiones del artículo se presentan en la sección VI.

II. Motivación

Generar pruebas criptográficas que permitan la verificación universal del proceso de mixing puede ser complejo, computacionalmente costoso, y puede poner en riesgo la privacidad de los votantes.

Algunos sistemas de mixing verificable [SK95], [FS01], [Ne01], consiguen una gran precisión (buena capacidad de detección de comportamientos maliciosos) a la vez que preservan la privacidad de los votantes, todo ello a costa de tener que realizar una gran cantidad de pruebas y verificaciones. Dado que estas pruebas y verificaciones suponen un gran coste computacional, estos sistemas resultan inadecuados para ser utilizados en elecciones reales con un gran número de votantes, ya que se ralentizaría el proceso excesivamente. Por esta razón, existen propuestas (por ejemplo en [BG02]) donde se utilizan estos sistemas para hacer un recuento paralelo de los votos al mismo tiempo que se usan métodos más rápidos (y menos precisos) para publicar los resultados provisionales de la elección.

Para mejorar la eficiencia del proceso de mixing, otros sistemas se enfocan en la reducción del coste de las verificaciones, sacrificando así en parte la privacidad de los votantes, o reduciendo la precisión del proceso de auditoría a un nivel aún suficiente para un proceso electoral. Por ejemplo, la propuesta de Random Partial Checking (RPC) [JJR02] sacrifica principalmente privacidad, mientras que la propuesta presentada en [Go02] preserva la privacidad de los votantes a costa de una reducción en la precisión y en la eficiencia, realizando más pruebas que ralentizan el proceso de verificación. En [BG02] se propone otro método que reduce la privacidad y precisión obtenidas a costa de la eficiencia, consiguiendo resultados que pueden resultar suficientemente buenos para un proceso de votación electrónica cuando se manejan grandes cantidades de votos.

El sistema de verificación de mixing que se presenta en este artículo es muy eficiente (comparable a las propuestas más rápidas) a la vez que preserva completamente la privacidad del votante y consigue un nivel alto de precisión.

III. Criptosistema

En nuestra propuesta, los votantes usan el criptosistema ElGamal (parametrizado apropiadamente para tener seguridad semántica [Pf94], [TY98]) para cifrar los votos. El criptosistema se compone de tres parámetros públicos: p, q, g; una clave pública h; y una clave privada x, definidos del siguiente modo:

- El módulo p es un primo seguro (safe prime), es decir, $p = 2q + 1$ y q es un número primo.
- g es un generador de G_q, el subgrupo de orden q de Z_p^*.
- La clave privada x es un elemento de Z_q, y la clave pública h se calcula como $h = g^x \ mod\ p$.

Para que los votos cifrados sean indistinguibles, las opciones de voto v se configuran de modo que todas provengan del conjunto de residuos cuadráticos, o del conjunto de residuos no cuadráticos modulo p. En el caso de que una opción de voto no encaje en el conjunto escogido se puede utilizar un relleno o padding.

Las opciones de voto se cifran utilizando exponentes aleatorios r pertenecientes a Z_q:

$c = (v \cdot h^r \ mod\ p, g^r \ mod\ p) = (c_1, c_2)$.

Así, una opción de voto se puede recuperar a partir del cifrado como $v = c_1 \cdot c_2^{-x} \ mod\ p$.

Este criptosistema tiene ciertas propiedades interesantes que utilizamos en nuestro proceso de verificación de mixing, tales como el recifrado y la operación homomórfica de los votos cifrados (que veremos en la sección IV-C).

III-A. Recifrado de los votos cifrados

Gracias a las propiedades del criptosistema ElGamal, un mensaje cifrado puede ser cifrado de nuevo (recifrado) usando un nuevo valor de aleatorización sin necesidad de modificar el proceso de descifrado: si el voto cifrado es $c = (v \cdot h^r \ mod\ p, g^r \ mod\ p) = (c_1, c_2)$, el recifrado se puede realizar como $c' = (c_1 \cdot h^{r'} \ mod\ p, c_2 \cdot g^{r'} \ mod\ p) = (v \cdot h^{r+r'} \ mod\ p, g^{r+r'} \ mod\ p) = (c'_1, c'_2)$, donde r' es un nuevo exponente aleatorio. El voto recifrado se puede descifrar como: $v = c'_1 \cdot c'^{-x}_2 \ mod\ p$.

IV. Proceso de mixing y verificación

IV-A. Breve descripción del método

El método de verificación universal para mixnets de recifrado que se presenta en este artículo combina las ventajas de la técnica RPC [JJR02] con las de la propuesta de *Optimistic Mixing* [Go02]: la revelación parcial de la información se combina con pruebas calculadas a partir de la operación homomórfica de grupos de votos, consiguiendo así mejores niveles de privacidad, robustez y precisión que estos métodos.

En un primer paso cada nodo de la mixnet mezcla y recifra los votos cifrados de entrada, guardando de forma secreta y segura los valores de permutación y recifrado aplicados a cada voto. Una vez que el último nodo ha mezclado y recifrado sus entradas, los votos, ya anonimizados, están listos para ser descifrados, pero antes de revelar su contenido se verifica que la mixnet ha realizado el proceso de forma correcta. En nuestro sistema suponemos que, en el caso de existir uno o más nodos maliciosos, éstos intentarán modificar el contenido de los votos que procesan para cambiar el resultado de la elección.

En el proceso de verificación los votos cifrados en la entrada de cada nodo se dividen en varios grupos independientes según una organización aleatoria propuesta por un verificador (o auditor). Esta organización en grupos se realiza al final del proceso de mixing, antes del descifrado, para evitar que los nodos puedan conocer información por avanzado que les permita falsear las pruebas. Entonces, cada probador - o nodo de la mixnet -, por turnos, proporciona al verificador la localización global a la salida del nodo de los votos pertenecientes a cada grupo definido en la entrada.

La localización global a la salida de un nodo de los votos pertenecientes a un determinado grupo no revela la posición exacta de cada voto individual. Por ejemplo, el nodo probador

podría revelar a qué grupo pertenecen los votos a la salida asignándoles un valor numérico.

Cuando el verificador divide los votos cifrados de entrada en grupos, multiplica los votos de cada grupo para obtener una *Prueba de Integridad Entrante* utilizando las propiedades homomórficas que se explican en la sección IV-C. Después de que el probador indique qué votos a la salida del nodo pertenecen a cada grupo definido en la entrada, el verificador puede multiplicar los votos que pertenecen a un mismo grupo para obtener una *Prueba de Integridad Saliente*. Para cada par PI Entrante/Saliente de cada nodo de la mixnet, el probador proporciona una prueba de conocimiento nulo para demostrar que la PI Saliente es el recifrado de la PI Entrante.

Dado que cualquier auditor puede calcular las pruebas de integridad y verificar las pruebas de conocimiento nulo, con este método conseguimos el objetivo de verificabilidad universal. Además, gracias a que se proporciona información sobre grupos y no sobre votos individuales, se mantiene la privacidad de los votantes.

En las próximas secciones se proporcionan detalles sobre como se generan los grupos, las pruebas de integridad y las pruebas de conocimiento nulo.

IV-B. Creación de los grupos

Cuando el proceso de verificación se inicia, el verificador define de forma aleatoria los grupos en los que se dividen los votos de entrada del primer nodo de la mixnet, enviando una matriz con los índices de las posiciones de los votos a agrupar. Dado que el tamaño de los grupos está predefinido (tal y como se explica al final de esta sección), la matriz tiene dimensiones gxn, donde n es el número de votos en cada grupo y g es el número de grupos.

Para $m = 6$ votos de entrada: $\{v_1, v_2, v_3, ..., v_6\}$; un número de grupos $g = 3$, y $n = 2$ votos por grupo, un ejemplo de matriz de agrupamiento podría ser:

$$\begin{pmatrix} v_3 & v_6 \\ v_1 & v_5 \\ v_2 & v_4 \end{pmatrix}$$

El probador, a su vez organiza, los votos de entrada siguiendo el orden marcado por la matriz de agrupamiento para definir los votos de cada grupo (en este ejemplo, los votos en las posiciones 3 y 6 pertenecerán al primer grupo). Utilizando la información de permutación del mixing, el probador indica al verificador, para cada voto a la salida del nodo, cuál es el grupo al que pertenece. Como la información se proporciona siguiendo el orden de las salidas del nodo de la mixnet, es imposible correlacionar individualmente los votos de entrada y salida (sólo el grupo al que pertenecen).

En los siguientes nodos de la mixnet los grupos de votos en la entrada se redefinen utilizando como referencia los grupos de salida del nodo previo. No se aconseja redefinir los grupos de forma aleatoria, dado que entonces un atacante podría analizar los votos pertenecientes a cada grupo en cada nodo y llegar a situar un voto en la salida de la mixnet en un grupo específico de votos en la entrada. De este modo se

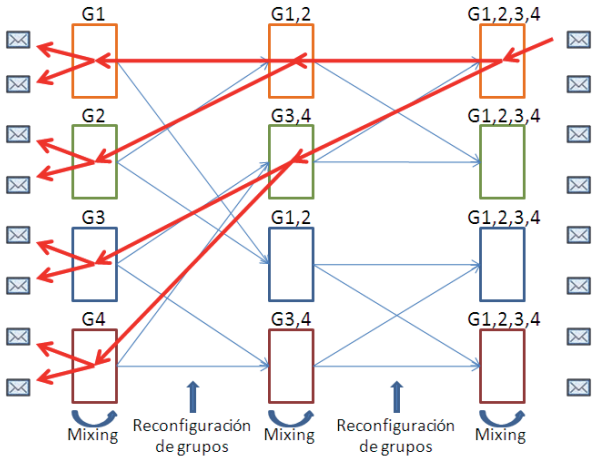


Figura 1. Trazabilidad de un mensaje en la mixnet.

conseguiría romper parcialmente la privacidad de los votantes, ya que la probabilidad de que un voto en la entrada de la mixnet estuviera en una posición cualquiera a su salida dejaría de ser $1/n$. Para prevenir este ataque, se recomienda definir los nuevos grupos de entrada en un nodo escogiendo votos pertenecientes a diferentes grupos de salida del nodo anterior, de forma que los grupos del último nodo de mixing estarán compuestos por al menos un voto perteneciente a cada uno de los grupos definidos en el primer nodo. De este modo, para un atacante será imposible escoger un voto en la salida de la mixnet y trazar su ruta a través de los nodos para encontrar el voto correspondiente en la entrada, ya que todos los votos en la entrada tendrán la misma probabilidad de corresponder al voto en la salida, tal y como muestra la figura 1.

Una propuesta para redefinir los grupos consiste en seleccionar votos pertenecientes a distintos grupos de salida del nodo anterior de forma consecutiva. En el ejemplo de la figura 1 el primer grupo del segundo nodo (G1,2) está formado por un voto del primer grupo del primer nodo (G1) y por uno del segundo (G2); el segundo grupo del segundo nodo (G3,4) está formado por un voto del tercer grupo del primer nodo (G3) y uno del cuarto (G4), y así sucesivamente.

El tamaño de los grupos es importante a la hora de mantener la privacidad del votante: si son demasiado pequeños, la distribución de votos/grupos a la salida de la mixnet puede no ser la aconsejada en el párrafo anterior (cada grupo en el último nodo no está formado por al menos un voto de cada uno de los grupos definidos en el primer nodo). Además, el tamaño también influye en la probabilidad de detectar votos manipulados durante el proceso de mixing: cuanto más pequeño es el grupo, más alta es la probabilidad de detección de manipulaciones. Así pues, los grupos deben definirse de una forma adecuada para conseguir el nivel más alto de detección sin comprometer la privacidad de los votantes.

Si t es el número de nodos de la mixnet (al menos dos) y m el total de los votos, el número mínimo de n votos que debe contener un grupo es:

$$n = \sqrt[t]{m} \quad (1)$$

Esta fórmula mantiene la privacidad total y optimiza la capacidad de detección de posibles manipulaciones. Como se muestra en la fórmula, en nuestra propuesta el número de nodos de la mixnet también contribuye a la precisión del sistema de verificación (permitiendo crear grupos más pequeños). No obstante, debe tenerse en cuenta que el hecho de añadir nodos reduce la eficiencia de la propuesta, ya que se incrementa el número de operaciones criptográficas a realizar durante los procesos de mixing y verificación.

IV-C. Generación de pruebas de conocimiento nulo a partir de las Pruebas de Integridad

La verificación de la integridad de los votos agrupados en cada nodo se basa en las propiedades homomórficas del criptosistema ElGamal: si tenemos dos votos v_1 y v_2, una operación de cifrado E y dos operaciones algebraicas ϕ y θ, un criptosistema con propiedades homomórficas se puede definir como aquél en el que $E(v_1)\ \phi\ E(v_2) = E(v_1\ \theta\ v_2)$.

Llamamos *Prueba de Integridad* la multiplicación de un grupo de votos. El resultado de multiplicar n votos de un mismo grupo a la entrada de un nodo de la mixnet, también llamada Prueba de Integridad Entrante, se puede definir como:

$$\prod_{i=1}^{n} c_i = (\prod_{i=1}^{n} v_i \cdot h^{r_i}, \prod_{i=1}^{n} g^{r_i}) = ((\prod_{i=1}^{n} v_i) \cdot h^{r_e}, g^{r_e}),$$

donde $r_e = \sum_{i=1}^{n} r_i$. La multiplicación del mismo grupo de votos a la salida del nodo (es decir, los mismos votos recifrados), se llama *Prueba de Integridad Saliente* y es igual a:

$$\prod_{i=1}^{n} c'_i = (\prod_{i=1}^{n} v_i \cdot h^{r_i + r'_i}, \prod_{i=1}^{n} g^{r_i + r'_i}) = ((\prod_{i=1}^{n} v_i) \cdot h^{r_e + r'_e}, g^{r_e + r'_e})$$

Como el nodo de la mixnet conoce los factores de recifrado aplicados a todos los votos, éste puede calcular la suma del factor de recifrado de los votos de un mismo grupo $\sum_{i=1}^{n} r'_i = r'_e$. Una vez obtenido el factor de recifrado conjunto, el nodo puede calcular una prueba de conocimiento nulo de recifrado no interactiva (NIZKP-RE) para probar que la Prueba de Integridad Saliente es el recifrado de la Prueba de Integridad Entrante (del mismo grupo) usando el factor de recifrado r'_e . Esta prueba puede basarse en el Protocolo de Identificación de Schnorr, com en [MA99], o en la prueba de Chaum-Pedersen de igualdad de logaritmos discretos [CP93].

De este modo, cualquier auditor puede calcular por él mismo las Pruebas de Integridad Entrante y Saliente de cada grupo de votos de un nodo y verificar la prueba NIZKP-RE para comprobar que las dos pruebas de integridad tienen los mismos contenidos. Dado que las pruebas de integridad están basadas en el producto homomórfico de los votos, aún existe la posibilidad de que un nodo malicioso pueda engañar al sistema. No obstante, gracias a la configuración de los grupos explicada en esta propuesta (ver sección IV-E1), la probabilidad de detección de una posible manipulación es bastante alta (por ejemplo, la probabilidad de detectar la manipulación de 2 votos en una elección con 10.000 votantes es del 99,91 %).

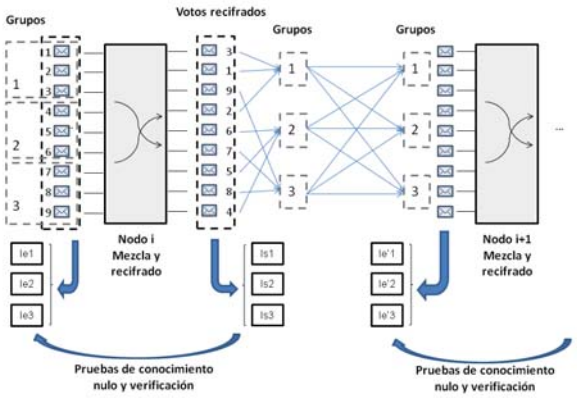


Figura 2. Verificación del proceso de mixing.

Las pruebas NIZKP-RE se realizan para todos los grupos de votos en cada nodo. Si estas pruebas se verifican de forma satisfactoria, se considera que la mixnet ha realizado el proceso de forma correcta.

IV-D. Resumen del protocolo de verificación

Para resumir, el protocolo de verificación implementa los siguientes pasos después del proceso de mixing:

1. Para el primer nodo de la mixnet, el verificador divide los votos de entrada en grupos aleatorios mediante una matriz de agrupamiento que envía al probador.
2. El verificador entonces calcula una Prueba de Integridad Entrante para cada grupo.
3. El verificador solicita al probador que indique la destinación de los votos pertenecientes a cada grupo en la salida del nodo y calcula una Prueba de Integridad Saliente para cada grupo.
4. El probador calcula una prueba de conocimiento nulo para cada grupo basada en el factor de recifrado de los votos en la salida del nodo. Con ellas demuestra al verificador que la Prueba de Integridad Saliente de cada grupo es el recifrado de la Prueba de Integridad Entrante del mismo.
5. En el siguiente nodo los grupos de entrada se redefinen de forma que cada grupo nuevo contiene votos de diferentes grupos de salida del nodo anterior. Los pasos 2-5 se repiten hasta que se verifica el correcto comportamiento del último nodo de la mixnet.

En la figura 2 se muestra un ejemplo del procedimiento de verificación y de la configuración de los grupos en cada nodo de la mixnet.

IV-E. Propiedades del método de verificación

IV-E1. Capacidad de detección: Dado que el proceso de verificación se basa en Pruebas de Integridad que se calculan

a partir de grupos de votos, un atacante podría aprovechar las propiedades homomórficas del algoritmo de cifrado para modificar votos en la mixnet sin ser detectado: en el caso de que las modificaciones se cancelaran durante el cálculo de las Pruebas de Integridad, éstas no serían detectadas durante el proceso de verificación. No obstante, como la configuración de los grupos es desconocida hasta que el proceso de mixing finaliza (después de que se hayan realizado las posibles modificaciones), la probabilidad de que un atacante modifique una cantidad de votos significativa sin ser detectado es negligible.

Esta probabilidad depende de la cantidad de votos en la mixnet, el número de grupos en los cuales se dividen los votos, y el número de votos manipulados (la probabilidad de que un atacante no sea detectado se reduce cuando aumenta el número de votos modificados). La probabilidad de detectar un par de votos manipulados es:

$$Psuccess = 1 - \frac{n-1}{m-1} \quad (2)$$

donde m es el número total de votos y n es el número de votos en cada grupo.

El par de votos manipulados no se detectaría en caso de que estuvieran en el mismo grupo y el valor de las modificaciones se cancelara con la multiplicación para calcular la prueba de integridad. Por esto, es importante mantener la relación adecuada entre el total de votos procesados por la mixnet y el tamaño de los grupos: cuanto más pequeños son los grupos, más alta es la probabilidad de que un atacante sea detectado (respetando el tamaño mínimo determinado por la ecuación 1 para preservar la privacidad de los votantes). En cambio, cuanto más grandes son los grupos, más rápido es el proceso de verificación.

La fórmula 1 proporciona esta relación: en una elección con 10.000 votos y una mixnet de dos nodos, el tamaño mínimo de los grupos preservando la privacidad de los votantes es de 100 votos. Con esta configuración, la detección de dos votos modificados es del 99 %. Si la mixnet tiene cuatro nodos, el tamaño mínimo de los grupos es de 10 votos y la probabilidad de detección aumenta al 99.91 %.

Las probabilidades de detectar un par de votos modificados en una mixnet compuesta por dos nodos (grupos más grandes) o por cuatro nodos (grupos más pequeños) se muestran en la figura 3. En los dos casos la probabilidad de detección tiende al 100 %, pero en el caso de tener más nodos esta probabilidad aumenta más rápidamente.

IV-E2. Eficiencia: Preservar la privacidad de los votantes dividiendo los votos en grupos no solapados afecta a la eficiencia de la mixnet. El coste computacional del método de verificación depende del número de votos en el sistema y de la cantidad de grupos creados para el proceso de verificación, ya que las pruebas se realizan sobre éstos. Para un mismo número de votos en la mixnet, cuanto más grupos se crean (más pequeños) más alta es la capacidad de detección, pero también se consumen más recursos.

En la figura 4 se muestra una comparación entre nuestro método y otros sistemas de verificación de mixnets en términos

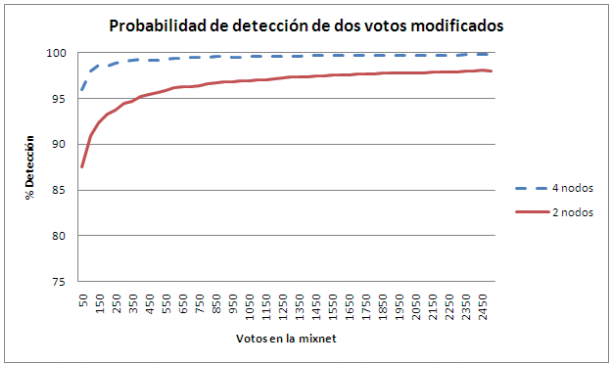


Figura 3. Probabilidad de detección en mixnets de dos y cuatro nodos.

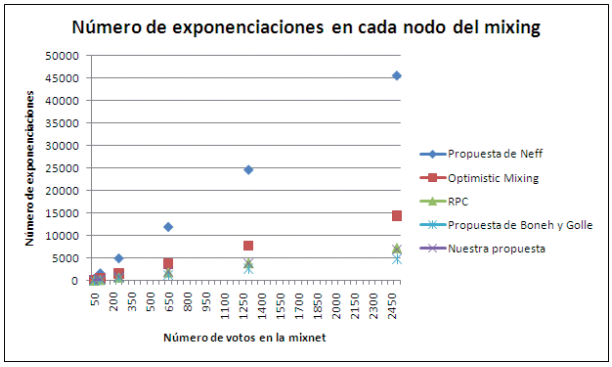


Figura 4. Coste de los diferentes sistemas de verificación de mixing.

de número de exponenciaciones necesarias (ya que éstas suelen ser las operaciones más costosas). Podemos comprobar que nuestro sistema es uno de los más rápidos cuando manejamos cantidades de votos razonablemente grandes.

V. Cifrado eficiente de los votos

Uno de los aspectos a tener en cuenta en la implementación práctica del protocolo es que los votos serán cifrados directamente con un algoritmo de clave pública (ElGamal). Esto puede significar un problema desde el punto de vista de rendimiento para el cifrado y descifrado de votos, sobre todo cuando la información de éstos supera la longitud del bloque de cifrado. Por ejemplo, en ElGamal la información que se puede cifrar en un solo bloque está limitada por las operaciones modulares que se realizan. El límite de tamaño de bloque coincide con la longitud de la clave pública, que hoy en día se recomienda que tenga unos 2048 bits. En este caso concreto, esto significa que podemos incluir un voto de hasta unos 292 caracteres (o 2048 bits) en un solo bloque cifrado. El exceso de caracteres se debería cifrar en un segundo bloque, lo que significaría realizar un segundo cifrado y por lo tanto otro par de exponenciaciones (una sobre la clave pública y otra sobre el generador). Generalmente, para estos casos, se combina el cifrado simétrico con el asimétrico (i.e., sobres digitales), puesto que el cifrado simétrico es menos costoso computacionalmente que uno asimétrico. El problema

es que esta combinación no es compatible con las propiedades homomórficas ni pruebas de conocimiento nulo requeridas por nuestra propuesta. Por lo tanto hay que buscar alternativas que permitan un cifrado más eficiente del voto cuando su información implique más de un bloque de cifrado.

Una opción consiste en comprimir los contenidos de los votos utilizando una representación de sus opciones mediante códigos alfanuméricos separados por caracteres especiales (p.e., un formato *comma-separated value* o CSV). Usando esta opción, se puede incluir en un único bloque cifrado un número mayor de opciones, pero existe una alternativa que además permite explotar las propiedades homomórficas del cifrado ElGamal. Esta consiste en representar las opciones de voto únicamente en un formato numérico.

Para la representación numérica del voto, utilizaremos números primos (o coprimos) prefijados, como en [Pe09]. Usando este formato, un voto contendría el producto de los números primos que representan las opciones de voto seleccionadas. Por ejemplo, si representamos una opción de voto con el número 2 y otra con el número 3, un voto en el que se hayan seleccionado estas dos opciones se representaría con el número 6 (i.e., el producto de 2 y 3). De este modo, como los números utilizados son primos, solo haría falta una factorización del descifrado del voto para obtener las opciones individuales seleccionadas originalmente. Esta representación además tiene la flexibilidad de poder cifrar inicialmente de forma individual las opciones de voto seleccionadas y luego obtener el voto mediante la operación de estas opciones cifradas. Esto es posible gracias a la propiedad homomórfica multiplicativa de ElGamal.

El cifrado individual de las opciones de voto implica a nivel de eficiencia un claro hándicap, pero lo consideramos porque es interesante desde el punto de vista de la auditoría de los votos, ya que permite verificar si un voto contiene opciones de voto válidas. Esta verificación se puede realizar mediante pruebas de conocimiento nulo que permitan verificar si un valor cifrado se encuentra dentro de un rango de valores prefijados, tal y como se explica en [CGS97]. De este modo, los votantes podrían enviar en lugar del voto cifrado, el conjunto de las opciones cifradas individualmente junto con las pruebas de conocimiento nulo de sus contenidos. Una vez verificado que los contenidos del voto son válidos, el voto se podría compactar mediante el producto de sus opciones cifradas. Esta compactación es importante para evitar problemas de rendimiento en la mixnet, ya que las operaciones de re-cifrado y las pruebas de integridad se continuarían haciendo sobre un único paquete cifrado (el voto compactado), en lugar de tantos paquetes como opciones de voto seleccionadas.

VI. Conclusiones

En este artículo hemos presentado un sistema de verificación de mixnets cuyo coste computacional es cercano al de la propuesta más eficiente (ver figura 4). En términos de privacidad, de las propuestas más eficientes, es la única que mantiene la privacidad total de los votantes. Además, el método consigue un buen nivel de precisión, con una probabilidad de detección de posibles manipulaciones cercana al 100 % (superior al 99 %) cuando se procesan más de 300 votos.

En resumen, comparada con los métodos actuales de verificación, nuestra solución es la más equilibrada en términos de eficiencia, privacidad y precisión, consiguiendo además la propiedad de verificabilidad universal.

Finalmente, dado que los sistemas de verificación de mixnets suelen requerir el uso de algoritmos de cifrado poco eficientes a la hora de cifrar mensajes relativamente largos (como ElGamal), también hemos propuesto un sistema en el que los votantes representan las opciones de voto en un formato numérico que aprovecha las propiedades homomórficas del algoritmo de cifrado para mejorar la verificabilidad del proceso y el rendimiento de la mixnet, así como del proceso de descifrado.

Referencias

[BG02] Boneh, D. and Golle, P. 2002. “Almost entirely correct mixing with applications to voting”, in *Proceedings of the 9th ACM Conference on Computer and Communications Security* (Washington, DC, USA, November 18 - 22, 2002). V. Atluri, Ed. CCS ’02. ACM, New York, NY, 68-77.

[CGS97] Cramer, R., Gennaro, R., Schoenmakers, B. “A secure and optimally efficient multi-authority election scheme”. In *EUROCRYPT ’97*. Lecture notes in computer science, vol. 1233. Springer, Berlin. pp. 103-118.

[Ch81] Chaum, D. L. 1981. “Untraceable electronic mail, return addresses, and digital pseudonyms”, in *Commun. ACM 24*, 2 (Feb. 1981), pp. 84-90.

[CP93] Chaum, D. and Pedersen, T. P. 1993. “Wallet Databases with Observers”, in *Proceedings of the 12th Annual international Cryptology Conference on Advances in Cryptology* (August 16 - 20, 1992). E. F. Brickell, Ed. Lecture Notes In Computer Science, vol. 740. Springer-Verlag, London, 89-105.

[FS01] Furukawa, J. and Sako, K. 2001. “An Efficient Scheme for Proving a Shuffle”, in *Proceedings of the 21st Annual international Cryptology Conference on Advances in Cryptology* (August 19 - 23, 2001). J. Kilian, Ed. Lecture Notes In Computer Science, vol. 2139. Springer-Verlag, London, 368-387.

[Go02] Golle, P., Zhong, S., Boneh, D., Jakobsson, M., and Juels, A. 2002. “Optimistic Mixing for Exit-Polls”, in *Proceedings of the 8th international Conference on the theory and Application of Cryptology and information Security: Advances in Cryptology* (December 01 - 05, 2002). Y. Zheng, Ed. Lecture Notes In Computer Science, vol. 2501. Springer-Verlag, London, 451-465.

[JJR02] Jakobsson, M., Juels, A., and Rivest, R. L. 2002. “Making Mix Nets Robust for Electronic Voting by Randomized Partial Checking”, in *Proceedings of the 11th USENIX Security Symposium* (August 05 - 09, 2002). D. Boneh, Ed. USENIX Security Symposium. USENIX Association, Berkeley, CA, 339-353.

[MA99] Markus, J. and Ari, J. 1999 “Millimix: Mixing in Small Batches. Technical Report.” UMI Order Number: 99-33., Certer for Discrete Mathematics & Theoretical Computer Science.

[Ne01] Neff, C. A. 2001. “A verifiable secret shuffle and its application to e-voting”, in *Proceedings of the 8th ACM Conference on Computer and Communications Security* (Philadelphia, PA, USA, November 05 - 08, 2001). P. Samarati, Ed. CCS ’01. ACM, New York, NY, 116-125.

[Pe09] Peng, K. 2009. “A Hybrid E-Voting Scheme”. In *Proceedings of the 5th international Conference on information Security Practice and Experience* (Xi’an, China, April 13 - 15, 2009).

[Pf94] Pfitzmann, B. “Breaking efficient anonymous channel”, in A. D. Santis, editor, *Advances in Cryptology (Eurocrypt ’94)*, volume 950 of LNCS, pages 332–340, Perugia, Italy, 9-12 May 1994. Springer-Verlag.

[SK95] Sako, K. and Kilian, J. “Receipt-free mix-type voting scheme - A practical solution to the implementation of a voting booth”, in *Advances in Cryptology -EUROCRYPT ’95*, Lecture Notes in Computer Science, Springer-Verlag, 1995.

[TY98] Tsiounis, Y. and Yung, M. 1998. “On the Security of ElGamal Based Encryption”, in *Proceedings of the First international Workshop on Practice and theory in Public Key Cryptography: Public Key Cryptography* (February 05 - 06, 1998).

Comparación de afinidades privada mediante isomorfismo de grafos

Juan Vera del Campo, Juan Hernández Serrano, Josep Pegueroles
Universitat Politècnica de Catalunya
Email: {juanvi,josep.pegueroles,jserrano}@entel.upc.es

***Abstract*—The next generation of social networks will provide advanced mechanisms to look for users that are similar and share interests. In order to achieve this, users should be able to calculate their affinity using a profile that describes them. But these profiles will include the interests, likes and dislikes of the users and then store very sensitive and private data. In this paper, we propose a protocol that lets two users that never met in advance to compare their profiles and check whether or not they are affine, without leaking any private information in case of mismatch. The protocol that we propose is based on the well known graph and subgraph isomorphism problems.**

***Index Terms*—social networks, profiling, privacy, security, graph isomorphism, subgraph isomorphism.**

I. Escenario y objetivos

En las redes sociales los usuarios publican sus puntos de vista, intereses y aficiones. Uno de los servicios que las redes sociales ofrecerán en un futuro inmediato es la búsqueda de personas que tengan intereses similares, con el objetivo de que se enlacen entre ellos y así se mejore el funcionamiento de la red. Pero para ello los usuarios deben publicar su perfil de forma que todo el mundo pueda compararse con él, lo que supone mostrar una gran cantidad de información personal que no todos los usuarios estarán dispuestos a publicar. Para proteger a estos usuarios, en este artículo proponemos un protocolo que compare a dos personas sin que sea necesario publicar ningún tipo de información privada, incluyendo sus intereses.

A. Modelo social

En la red social estudiada, dos usuarios con perfiles similares querrán enlazarse entre sí porque cada uno de ellos tendrá documentos que pueden ser de interés al otro. Así, los perfiles de un usuario de deben crear a partir de los perfiles de los documentos que comparte dentro de la red. A continuación veremos cómo es posible.

En una red P2P hay un conjunto de documentos únicos $R = \{r_1, r_2, \ldots, r_m\}$. Estos documentos o recursos pueden describirse con meta-datos, campos e inspección interna de forma que es posible definir conjuntos de palabras, *bag-of-words* [1]. De acuerdo con [2], es posible definir una ontología de clasificación de los recursos en R. Esto es, se obtienen n categorías semánticas con las que podemos describir los recursos en R. Así, cada recurso $r_i \in R$ tiene asociado un perfil dentro de esa ontología $\bar{p}(r_i) = (c_1, c_2, \ldots, c_n)$. En este trabajo supondremos que los componentes c_i de $\bar{p}(r)$ son números reales entre 0 y 1 y que cada uno de las categorías es independiente de las demás. Llamamos **espacio social** $\mathbb{P} = [0, 1]^n$ al conjunto de todos los perfiles posibles. Sobre este conjunto definimos una función distancia $d : \mathbb{P} \times \mathbb{P} \to \mathbb{R}$.

A partir de los perfiles de los documentos que un usuario comparte es posible crear un perfil de usuario. Un ejemplo trivial es asociar como perfil de un usuario la media de los perfiles de los documentos que comparte dentro de la red. Otra posibilidad más compleja es que se tenga en cuenta el comportamiento pasado del usuario así como la descripción que haga él mismo de sus intereses a la hora de calcular el perfil de un usuario. Sea como sea, asumimos que los perfiles de usuario también pertenecerán al espacio social $\mathbb{P}$. Así, dado un número real $\lambda \in \mathbb{R}$, decimos que dos usuarios a y b son **afines** si y solo si sus perfiles asociados verifican $d(A, B) < \lambda$.

En este trabajo no hacemos ninguna suposición sobre cómo los perfiles en $\mathbb{P}$ son asignados a cada documento, ni hacemos ninguna asunción sobre la distancia particular usada en d. Pero aunque en este trabajo estemos usando perfiles abstractos, creemos que en un entorno real que dos perfiles sean afines captura la similitud entre usuarios de una red social. En este sentido, si dos usuarios son afines, entonces sus gustos son similares y probablemente estos dos usuarios querrán conocerse, conectarse o compartir conocimientos y documentos.

B. Modelo de seguridad

En este trabajo asumimos que los usuarios no están organizados de ninguna manera ni se conocen de antemano. Además, no existe un servidor central que pueda manejar la confianza entre los usuarios, ni autentificar las identidades de los mismos. Es decir, no es posible para un usuario comprobar si el perfil presentado por otro usuario que resulta ser afín no es falso. Esto es, un usuario malicioso podría falsear su perfil o incluso auto-asignarse cualquier perfil posible dentro de $\mathbb{P}$, e incluso cambiar su perfil de forma dinámica y aparecer con diferentes identidades. Finalmente, no exigiremos que existan canales seguros entre los nodos de la red, así que la comunicación entre dos usuarios puede ser perfectamente conocida por un usuario malicioso que la está interviniendo, ya sea pasivamente o como *man-in-the-middle*.

Los atacantes de este sistema son de dos tipos. El primero de ellos es un observador de la comunicación pasivo que no participa en el protocolo pero que tiene la posibilidad de obtener y analizar todos los mensajes intercambiados por Alice y Bob. Este atacante gana si es capaz de obtener cualquier

información de cualquiera de los perfiles de Alice (A) o Bob (B). Este requisito se verifica no solo si el atacante es capaz de descubrir A ó B, ya sea completos o parcialmente, sino también se verificará si es capaz de descubrir la afinidad $d(A, B)$ o incluso si es verdad que $d(A, B) < \lambda$[1].

El siguiente tipo de atacante es un usuario activo Malloy que corre el protocolo con cualquier perfil M que le pueda interesar ejecutar. Fácilmente se comprueba que si Malloy se asigna a sí mismo un perfil afín con Alice, entonces Malloy conoce que A está dentro de la hiperesfera centrada en M y de radio λ. Si este radio es muy pequeño, conocerá el perfil de Alice con muy poco error. El objetivo de este trabajo es que la probabilidad de que Malloy se asigne un perfil a sí mismo que sea afín con Alice no es mayor que la simple suerte, con lo que el ataque se vería reducido a una forma de ataque por fuerza bruta. Además, si el perfil M no es afín, Malloy no debería obtener mayor información sobre la distancia que le separa realmente de Alice, de forma que no sea posible ningún tipo de ataque por triangulación de perfiles.

C. Objetivo

Dos usuarios diferentes, Alice y Bob, se encuentran en una red social por primera vez. Quieren conocer si es verdad o no que son afines. Formalmente, si A es el perfil de Alice y B es el perfil de Bob, quieren conocer si es cierto o no que la distancia entre ambos es menor que λ, para un cierto valor $\lambda \in \mathbb{R}$ que se ha acordado previamente:

$$1 : \{d(A, B) < \lambda\} \tag{1}$$

La salida de (1) es 1 si se verifica la desigualdad, o 0 si no lo hace. Alice y Bob no desean que la otra parte gane ningún tipo de conocimiento aparte del resultado de la comparación. Es decir, no solo el perfil real de cada uno de ellos sino también la distancia que los separa debe ser mantenida en secreto.

En este trabajo definiremos un protocolo que permita a dos usuarios Alice y Bob validar si son afines para un cierto λ sin dar más información que la salida de (1). Este protocolo estará basado en los problemas del isomorfismo de grafos y subgrafos.

D. Una vista más amplia: caso de uso

Pretendemos utilizar el protocolo que describiremos en este trabajo de la siguiente forma. En una red social los usuarios se encuentran con otros muchos usuarios. Queremos diseñar un sistema que permita enlazar a los usuarios de la red social que compartan intereses comunes. De esta manera, podrán fraternizar, compartir, o realizarse recomendaciones unos a los otros sobre documentos que cada uno de ellos posea. Si un usuario se enlaza con otros usuarios con sus mismos intereses dentro de la red social, entonces los documentos de estos usuarios probablemente también sean de interés para el primero de ellos.

Pero los usuarios maliciosos están por todas partes. Redes como Facebook o LinkedIn ganan dinero con la información privada de los usuarios. Google guarda información personal sobre sus usuarios que permite la creación de perfiles enormemente detallados. Empresas de espionaje industrial, personal, comerciales, otro usuarios particulares o gobiernos podrían utilizar la información personal de los usuarios en su contra.

Para resolver estos ataques a la privacidad de un usuario de redes sociales, apostamos por la creación de una red descentralizada donde los usuarios puedan conocer a otros con sus mismos intereses sin necesidad de enviar sus perfiles a un servidor central controlado externamente, ni que otros usuarios maliciosos puedan recoger una gran cantidad de perfiles de los demás usuarios de la red.

II. Trabajo relacionado

A. Computación multiparte segura

Hay varias propuestas en la literatura que intentan resolver un problema similar mediante la utilización de computación multiparte segura. En teoría, si cada una de las partes corre un protocolo de computación multiparte usando su propio perfil como entrada, podría calcularse la distancia entre ambos perfiles sin necesidad de que los mismos se intercambien. Incluso es posible definir una función que no calcule la distancia, sino que resuelva directamente (1). [4] describe una puerta condicional que es capaz de resolver operaciones complejas en computación multiparte. Uno de los ejemplos de aplicación es precisamente la resolución de una función muy similar a (1). Pero los autores limitan la definición de intereses a un número binario $0-1$, y la métrica a la distancia de Hamming entre los perfiles. Intentar aplicar esta misma solución al entorno más complejo descrito en la sección I-D, con vectores de centenares de categorías cada una cuantificada en 8 bits, llevaría a un número necesario de mensajes que los usuarios tienen que intercambiar totalmente inaceptable. Los autores pensamos que el número de interacciones necesarias si se utilizan sistemas de computación multiparte limita seriamente su aplicación para el cálculo de afinidad entre perfiles a escenarios muy particulares y simplificados.

El problema del isomorfismo de grafos se ha utilizado en ocasiones en la literatura de seguridad. [5] presenta un protocolo zero-knowledge genérico que puede utilizar cualquier problema *NP-Complete* para autentificar a un usuario, y dos de los ejemplos explícitamente incluidos son el problema de isomorfismo de grafos y subgrafos. Aunque el protocolo genérico no es directamente aplicable en nuestro escenario, lo hemos utilizado como inspiración de nuestro trabajo. Por otro lado, [5] no toma en cuenta las aproximaciones a la solución del problema de isomorfismo, que analizaremos en la sección V. [6] describe un protocolo para realizar autenticación mutua entre cliente y punto de acceso a una red WiFi aprovechando la complejidad del problema de isomorfismo de grafos. De nuevo el protocolo presentado allí no es de aplicación en nuestro escenario de perfiles sociales.

[1] Al menos, durante la ejecución del protocolo. Si Alice y Bob continúan comunicándose con posterioridad, entonces un atacante podría inferir que es muy probable tanga perfiles afines. Protegerse de este ataque está fuera de los objetivos de este trabajo, aunque podría solucionarse si se exige que las comunicaciones siguientes entre Alice y Bob se realicen a través de sistemas que garanticen el anonimato como [3]

B. Teoría de grafos

Un grafo es un par $G = \{V, L\}$ formado por un conjunto de vértices $V = \{v_1, v_2, ..., v_n\}$ y un conjunto de enlaces entre ellos $L = \{(v_i, v_j), (v_l, v_k), ...\}$. En este trabajo solo consideraremos grafos simétricos, es decir, los enlaces en L no tienen dirección y no hay enlaces en bucle que empiecen y acaben en el mismo nodo. Un subgrafo $G_s = \{V_s, L_s\}$ de G es un grafo tal que $V_s \subset V$ y $L_s = L \cap (V_s \times V_s)$. Es decir, el subgrafo está formado por un subconjunto de vértices del grafo original y todos los enlaces que unen a este subconjunto de vértices del grafo original. Dos grafos G y H son isomórficos si y solo si comparten el mismo conjunto de vértices V y existe una permutación $\pi : \{1, 2, ..., n\} \rightarrow \{1, 2, ..., n\}$ tal que $\forall\ i, j (v_i, v_j) \in L_G \leftrightarrow (v_{\pi(i)}, v_{\pi(j)}) \in L_H$. Es decir, si podemos reetiquetar los vértices de G de forma que tengamos el grafo H. En este caso, escribimos $H = \pi G$.

En teoría de la complejidad, se dice que un problema pertenece al conjunto NP si existe una máquina de Turing no determinista (NDTM) que pueda resolver el problema en un tiempo polinomial. Una definición ligeramente diferente pero equivalente es que NP alberga la clase de problemas en las cuales una solución puede probarse en un tiempo polinomial por una máquina de Turing determinista (DTM). Un subconjunto interesante de NP es el conjunto $NP - Complete$. Un problema p es $NP - Complete$ si es NP y cualquier otro problema NP puede ser convertido en p en un tiempo polinomial. Actualmente, los problemas en $NP - Complete$ se consideran intratables, ya que no hay ningún algoritmo conocido para una máquina DTM que pueda resolver un problema $NP - Complete$ en un tiempo polinomial. En lo que nos interesa, el problema de comprobar si es verdad o no que un grafo G es isomórfico con un subgrafo de H es un problema $NP - Complete$, mientras que el problema de comprobar si dos grafos G y H son isomorfos es un problema NP (pero no se sabe si $NP - Complete$ ó P).

III. Descripción del protocolo

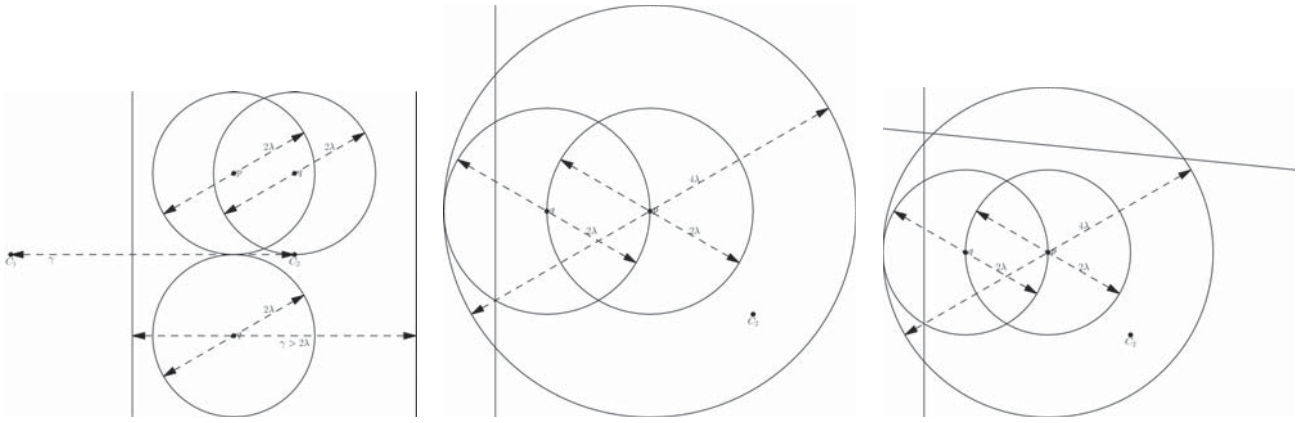
A continuación detallamos el protocolo propuesto para alcanzar los objetivos de la sección I-C. Dados dos usuarios, a y b, con perfiles A y B, se ponen de acuerdo en los números reales γ y λ. λ será el utilizado para comprobar si se verifica (1), mientras que estudiaremos las restricciones sobre γ en la sección III-A. Ambos usuarios ejecutan el siguiente protocolo.

1) Ambos usuarios a y b se ponen de acuerdo en un grafo $G = (V, L)$, construido de tal manera que: 1. todos sus vértices $V = \{v_1, v_2, ..., v_g\}$ están separados al menos una distancia γ; 2. los enlaces en L son aleatorios. G es la entrada común al protocolo, y es conocido por ambos usuarios. El número de vértices en G se representará como g. Vea la sección V para requisitos adicionales para este grafo.
2) A cada punto del espacio social $\mathbb{P}$ se le asigna el vértice de V que esté más cercano. Es decir, se define una función de clusterización $c : \mathbb{P} \rightarrow V$ de tal forma que el vértice v_i asignado a un punto p del espacio sea el que minimice $d(v_i, p)$.
3) Ambos usuarios a y b se ponen de acuerdo en un número real $\mu > \gamma$.
4) De forma secreta, el usuario, $x \in \{a, b\}$ calcula los conjuntos $HS(X, \lambda)$ y $HS(X, 2\lambda)$, siendo el conjunto $HS(P, r) = \{Z \in \mathbb{P} | d(P, Z) < r\}$. Es decir, la hiperesfera $HS(A, \lambda)$ es el conjunto de perfiles de $\mathbb{P}$ que están dentro de un radio de λ unidades del perfil A.
5) De forma secreta, el usuario x calcula los conjuntos C_x y C_x^2, ambos subconjuntos de V. El subconjunto de vértices $c(HS(P, r)) \subset V$ está formando por los vértices de V asignados a los puntos de la hiperesfera $HS(P, r)$ a través de la función c. Abusando de la notación, escribiremos $C_x = c(HS(X, \lambda))$ y $C_x^2 = c(HS(X, 2\lambda))$.
6) Cada usuario calcula los grafos H_x y H_x^2 como los subgrafos de G derivados de los subconjuntos de vértices C_x y C_x^2 extendidos a una distancia μ. Esto es, H_x incluye los vértices C_x y todos aquellos vértices de V que estén a una distancia menor que μ de cualquier vértice de C_x. H_x^2 se construye de forma similar, utilizando C_x^2 como conjunto de vértices base.
7) Cada usuario x escoge un isomorfismo cualquiera $\phi_x : V \rightarrow V$, y envía al otro participante $\phi_x(H_x)$ y $\phi_x(H_x^2)$.
8) Si cualquiera de los pares $(\phi_y(H_y), H_x)$, $(\phi_y(H_y^2), H_x)$ ó $(\phi_y(H_y), H_x^2)$ es isomorfo, cada usuario acepta individualmente que $d(A, B) < \lambda$.

Análisis: El núcleo de este protocolo es que el único dato que se intercambian los usuarios son los grafos $\phi_x(H_x)$ y $\phi_x(H_x^2)$. Estos conjuntos se crean a través de isomorfismos sobre los subgrafos de G formado con los vértices cuyos perfiles asociados son los más cercanos al perfil de x. Si un atacante obtiene el grafo $\phi_x(H_x)$ y es capaz de deshacer el isomorfismo, entonces será capaz de obtener este subgrafo original y podrá hacer una estimación bastante aproximada del perfil del usuario. Pero para ello tendría que resolver el problema del isomorfismo de subgrafos, que como se ha discutido en la sección II-B, es un problema $NP - Complete$ y actualmente considerado intratable. Aún así, dos usuarios legítimos tendrán que resolver el problema del isomorfismo de grafos en el último paso del protocolo. Este es un problema NP, pero no se sabe si es $NP - Complete$. En la sección V se discutirá sobre la viabilidad computacional de los usuarios para resolver este problema. En nuestro caso, el parámetro μ sirve para modular el número de vértices que tendrá el grafo $\phi_x(H_x)$ final, llevando el escenario a la zona donde el problema de isomorfismo de grafos es resoluble, pero no lo es el problema del isomorfismo de subgrafos.

A. Límites para el parámetro γ

A continuación estudiaremos los conjuntos C_x y C_x^2. La idea sobre la que soportamos nuestro protocolo es que dos usuarios cuyos perfiles verifiquen (1) deberían tener alguno de estos conjuntos iguales. Como veremos, la creación de estos conjuntos nos limitará los posibles valores del parámetro γ. Estudiaremos varios casos de intersecciones entre las hiperesferas HS_x y los clusters creados por la función c:



(a) Centro del clúster (b) Borde del clúster (c) Esquinas del clúster

Figure 1. Los tres escenarios analizados de cortes de clusters e hiperesferas

a) Zona central de los clusters: Si una hiperesfera $HS(p, \lambda)$ es mayor que un clúster, entonces ésta siempre contendrá más de un clúster y la comparación de $C(p, \lambda)$ y $C(q, \lambda)$ pierde sentido en el caso general. Para evitarlo, se exigirá que la distancia mínima entre dos vértices de G sea $\gamma_{min} = 2\lambda$. En este caso una hiperesfera perfectamente centrada en el clúster no cortará a ningún clúster más, y por tanto es posible comparar los conjuntos C_x de dos usuarios. Por otro lado, si $\gamma \gg 2\lambda$, entonces un solo clúster podría albergar dentro de sí mismo varias hiperesferas que tengan perfiles más lejanos que λ, como se muestra en la figura 1(a), resultando en un aumento de los falsos positivos. Así, conviene acotar un γ_{max} para reducir el número de falsos positivos.

b) Bordes: En los bordes de los clusters puede darse el caso de que una hiperesfera centrada en p esté contenida en un solo clúster mientras que otra centrada en q tal que $d(p, q) < \lambda$ corte a dos o más clusters adicionales. En este escenario, simplemente comparar los dos conjuntos de clusters llevaría a un falso negativo. Por tanto, se ha introducido en el protocolo un nuevo conjunto de clusters C_x^2, los cortados por una hiperesfera centrada en p y con radio 2λ. Efectivamente, esta hiperesfera contiene a $HS(q, \lambda)$ y por tanto todos los clusters cortados por $HS(q, \lambda)$ también serán cortados por $HS(p, 2\lambda)$. Este es el motivo por el que el protocolo comprueba durante la fase 8 no solo el corte de las hiperesferas de radio λ, sino también el corte de las hiperesferas de radio una λ y otra 2λ. El lector podrá comprobar que no se gana nada comparando los cortes de las hiperesferas 2λ, ya que sería exactamente el mismo caso con los mismos errores que comparar las hiperesferas con radios λ. Con esta modificación, el protocolo resultará positivo si cualquiera de estos conjuntos de clusters comparados son iguales. De igual forma y como muestra la figura 1(b), para que dos hiperesferas de radio 2λ no corten clusters diferentes, es necesario que $\gamma > 4\lambda$ para que dos bordes estén separados al menos la distancia equivalente a un diámetro de la hiperesfera.

c) Esquinas: Es un hecho conocido que las únicas figuras que mantienen una distancia fija llamada diámetro entre dos puntos de sus bordes son el triángulo equilátero en espacios bidimensionales y las hiperesferas en cualquier dimensión. Además, no es posible hacer una partición del espacio solo con hiperesferas. De esta forma, la implicación descrita en el párrafo anterior solo funciona en un sentido y aunque los clusters cubiertos por la hiperesfera $HS(q, \lambda)$ también estén cubiertos por la hiperesfera $HS(p, 2\lambda)$, esta última también puede cubrir nuevos clusters. En este caso, los conjuntos de clusters C_x y C_x^2 son diferentes, y por tanto el protocolo dará un falso negativo. Llamamos "esquinas" a las zonas del clúster donde no es posible mantener la restricción sobre γ, como muestra la figura 1(c). No hemos sido capaces de encontrar una solución sencilla a este problema sin añadir complejidad adicional al protocolo, aunque se puede minimizar su efecto sobre los falsos negativos maximizando el valor de γ. En la sección de simulación estudiaremos hasta qué punto influye en los resultados obtenidos.

B. Límites en el valor λ

Para esta sección utilizaremos la métrica euclídea para calcular distancias entre perfiles de usuario. Aunque los resultados numéricos no coincidan si utilizamos otras métricas, los resultados cualitativos sí que lo harán.

Si asumimos que los usuarios se extienden uniformemente en el espacio social, el perfil medio que podemos encontrar es $p = \{\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2}\}$, y en este caso, la distancia media máxima a cualquier otro perfil en $\mathbb{P}$ es:

$$\bar{d}_{max} = \sqrt{\sum_n (\frac{1}{2})^2} = \frac{\sqrt{n}}{2} \tag{2}$$

Según los límites encontrados en la sección anterior para los elementos de G, la distancia máxima entre dos perfiles cualesquiera g_i y g_j de G es $\gamma_{max} = 4\lambda$, así que:

$$4\lambda < d(g_i, g_j) < \bar{d}_{max} = \frac{\sqrt{n}}{2} \tag{3}$$

$$\lambda < \frac{\sqrt{n}}{8} \tag{4}$$

La ecuación (4) da un valor máximo para el umbral que se puede utilizar en una dimensión n para comparar perfiles. Con valores de λ por encima de este valor máximo, no será posible

crear los parámetros necesarios para correr el protocolo. Por ejemplo, en una red social donde se definen perfiles con $n = 200$ categorías, el valor umbral máximo que pueden usar los usuarios para compararse con este protocolo es $\lambda = 1,178$.

IV. Simulación del protocolo

En esta sección simularemos el protocolo que ha sido definido en este documento. Para ello definiremos un espacio social de $n = 20$ categorías. Utilizaremos $\gamma = 0.1$ y definiremos un grafo G con los vértices en cada uno de los puntos separados γ en cada dirección. Para simplificar las simulaciones pero sin pérdida de generalidad, ya que solo pretendemos comprobar la validez del protocolo, definiremos los clusters como los hipercubos centrados en cada uno de los vértices de G de arista γ. Finalmente, para comprobar la validez del protocolo bastará con comprobar si coinciden los conjuntos C_x ó C_x^2 de cada usuario.

En la figura 2(a) se muestra la relación de falsos negativos con respecto a los positivos, y de falsos positivos con respecto a los negativos para varios valores de γ/λ. Del apartado anterior concluimos que $\gamma > 4\lambda$ y a la vez tan pequeño como sea posible. Por inspección de la figura 2(a), un valor de $\gamma = 6\lambda$ consigue un ratio de falsos negativos cercano al 20%, y este ratio disminuye exponencialmente con la relación γ/λ. A la vez, el ratio de falsos positivos con respecto a los negativos aumenta lentamente con la relación γ/λ, siendo cercano al 40% para el valor $\gamma = 6\lambda$.

Pero más importante aún es la comparación de la figura 2(b). Esta figura muestra la media de distancias efectiva de los perfiles que el protocolo considera como positivos con respecto al umbral λ. Es decir, un valor de este parámetro de 2 significa que el protocolo considera positivos los perfiles con distancia media 2λ. Como se comprueba en la figura, las simulaciones muestran que esta distancia efectiva aumenta de forma aproximadamente lineal a medida que aumenta el ratio γ/λ. Así, queda justificado el análisis de la sección anterior, recomendando un valor de γ tan pequeño como sea posible. Para $\gamma = 4\lambda$, que cumple los requisitos del protocolo, la distancia efectiva de los perfiles positivos es 2λ. Aunque este valor de falsos positivos puede parecer alto, el lector recordará que estos falsos positivos se darán entre aquellos perfiles que compartan clúster con el perfil de ensayo. Esto es, la distancia máxima entre el perfil a considerar y cualquiera de los falsos positivos tendrá que ser menor que γ. Así, el protocolo en realidad es capaz de comparar perfiles no con el umbral λ, sino con un umbral λ' intermedio entre λ y $\gamma = 4\lambda$.

Finalmente, el caso de uso propuesto en la sección I-D se ha estudiado en otros trabajos como [7]. En el algoritmo de enrutamiento allí, los falsos positivos ayudan en la creación de grupos de afinidades entre usuarios. En ese caso, un falso positivo que además tiene la distancia máxima limitada con γ no solo no es un inconveniente para el sistema, sino que es una ayuda para el mismo.

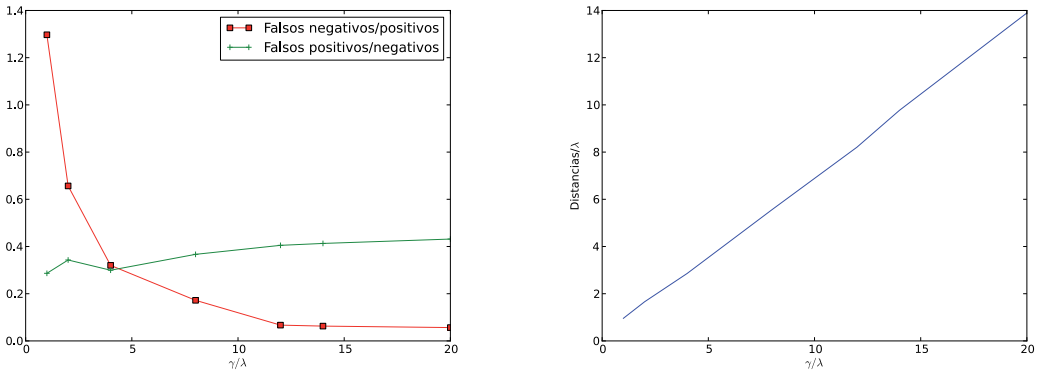
V. Discusión: viabilidad computacional

El reconocimiento de formas en grafos es un campo con 30 años de investigación ininterrumpida. Aunque el problema de isomorfismo de subgrafos sea $NP - Complete$ y no se conozca la clasificación final del problema de isomorfismo de grafos, se han presentado algoritmos capaces de resolver en algunos casos ambos problemas en un tiempo razonable. De hecho, para que nuestra propuesta sea viable es necesario que se cumplan dos condiciones: 1.- que dos usuarios puedan decidir rápidamente si los grafos finales resultantes de la ejecución del protocolo son isomorfos 2.- que dado uno de estos grafos, no se pueda encontrar a qué parte del grafo original G corresponde. La primera condición supone que los grafos finales tienen que ser tales que el problema de decisión de isomorfismo de grafos sea tratable, mientras que la segunda exige que el problema de búsqueda derivado del problema de decisión de isomorfismo de subgrafos no sea computable.

En [8] se presenta un algoritmo para la resolución de ambos problemas. Los autores defienden que su propuesta tiene complejidad temporal $O(N^2)$ en el mejor de los casos, y $O(N!N)$ en el peor, siendo N el número de nodos en el grafo. Además, presentan resultados de la ejecución del algoritmo para el reconocimiento de grafos isomorfos, pudiendo resolver el problema para $N = 1000$ nodos en tiempos cercanos al segundo. Además, en este mismo trabajo se presentan los resultados obtenidos con otros algoritmos clásicos, que obtienen una complejidad temporal similar en algunos casos. Así, podemos concluir que para grafos con un número de vértices cercano a 1000 el problema del isomorfismo de grafos es resoluble en un tiempo perfectamente razonable, y podemos asumir que el primer requisito de nuestra propuesta se puede alcanzar si exigimos que los grafos C_x y C_x^2 tienen un número de nodos como máximo de 1000 nodos.

Analizaremos a continuación la viabilidad del segundo requisito, que el problema del isomorfismo de subgrafos no sea resoluble y por tanto un atacante no pueda identificar un subgrafo de G que sea isomorfo con C_a, porque en ese caso conocería los nodos de G afines a a. Como se describió en la sección I-D, queremos aplicar este protocolo en un sistema de recomendación. Para que el sistema de recomendación sea útil se debe utilizar un número alto de categorías. Consideramos, así, usuarios que muestran su interés en cientos de categorías diferentes y hemos trabajado en espacios de $n = 200$ categorías. En este caso, incluso suponiendo un valor alto para la distancia mínima entre perfiles dentro de G como $\gamma = 0.2$, el número de posibles vértices dentro de G es enorme, $N_{max} = \frac{1}{\gamma}^n = 5^{200}$. Un número tan grande no puede ser tratado en un tiempo razonable por los algoritmos clásicos ni por las nuevas propuestas como [8], así que podemos considerar que en el caso general de G y para nuestro escenario, los dos requisitos de viabilidad computacional se verifican.

Pero existe una línea de ataque descrita en la literatura actual de privacidad que podría aplicarse en este caso. Si el atacante es capaz de introducir dentro de G un subgrafo H que cumpla ciertas características, la resolución del problema puede simplificarse. Por ejemplo, en [9] se presenta un método para introducir un pequeño grafo H dentro de otro mucho mayor G, y que pueda identificarse después de isomorfismos. Si un atacante puede introducir estos subgrafos durante la creación de G, podría de forma efectiva controlar varias zonas



(a) Falsos positivos y negativos en relación a γ/λ

(b) Distancias efectivas que el protocolo identifica como afines

Figure 2. Resultados de la simulación para $\gamma = 0, 1$, $n = 100$

del espacio social. Como los autores demuestran en su trabajo, para que este ataque tenga éxito, el grafo H introducido por el atacante tiene que tener como máximo del orden de $O(\log(N))$ nodos, y cada nodo del orden de $O(\log(N))$ vecinos. Los autores son capaces de recoger información con un subgrafo tan pequeño como 7 nodos introducido en una red social de 4 millones de nodos. Estos subgrafos H no tienen ninguna característica especial que permita a personas diferentes al atacante identificarlos, con lo que pueden permanecer perfectamente ocultos dentro de G.

Para resolver este problema proponemos que la creación de G sea controlada por ambas partes. Si G se crea dinámicamente en el momento de la comparación, se debe utilizar un algoritmo que impida la introducción de los grafos H arbitrarios. Por ejemplo, si ambas partes acuerdan un algoritmo de generación de números aleatorios y una semilla, y utilizan este algoritmo para la creación de enlaces aleatorios dentro de G, no es posible inyectar grafos H en G. Otra opción es que ambas partes se aseguren de que el grafo G verifica la propiedad de "k-anonimato de vecinos", es decir, que dado cualquier nodo y su vecindario, se puedan encontrar k vértices en la red cuyo vecindario es isomorfo al nodo que estamos testeando. En [10] puede encontrarse un algoritmo de generación de un grafo G que cumple esta propiedad.

VI. Conclusiones y líneas futuras

En este documento hemos desarrollado un protocolo para el cálculo de afinidades entre dos usuarios sin que cada uno de ellos aprenda más información sobre el otro que la afinidad. La base del protocolo es el problema de isomorfismo de grafos, que se considera en la actualidad no tratable en general aunque se hayan hecho aproximaciones en su solución.

Mediante simulaciones constatamos que el protocolo tiene un éxito parcial, y aunque es posible limitar el número de falsos negativos tanto como sea necesario, el número de falsos positivos es creciente con γ. Aún así, ya que conviene tener un γ lo más pequeño posible y en cualquier caso los perfiles positivos tendrán una distancia menor que γ entre sí, las consecuencias de los falsos positivos son limitadas.

Para comprobar las características estadísticas del protocolo hemos realizado algunas simulaciones en un escenario simplificado. En particular, hemos considerado particiones del espacio social como hipercubos. Existen particiones más óptimas que no se han considerado, y creemos que es posible definir dinámicamente el grafo G de forma que minimice los falsos positivos. Estas son líneas abiertas de investigación para trabajos futuros.

Agradecimientos

Este trabajo ha sido financiado en parte con el soporte del programa Consolider-Ingenio 2010 (ARES, CSD 2007-0004) y CICYT SECONNET (TSI2005-07293-C02-01).

References

[1] C. D. Manning, P. Raghadan, and H. Schütze, *An Introduction to Information Retrieval*. Cambridge University Press, April 2009.

[2] D. T. Tran, S. Bloehdorn, P. Cimiano, and P. Haase, "Expressive resource descriptions for ontology-based information retrieval," in *Proceedings of the 1st International Conference on the Theory of Information Retrieval (ICTIR'07), 18th - 20th October 2007, Budapest, Hungary*, 2007, pp. 55–68. [Online]. Available: http://www.aifb.uni-karlsruhe.de/WBS/pha/publications/ontology-ir-ictir07.pdf

[3] M. Reiter and A. Rubin, "Crowds: Anonymity for web transactions," *ACM Transactions on Information and System Security*, vol. 1, no. 1, June 1998.

[4] B. Schoenmakers and P. Tuyls, "Practical two-party computation based on the conditional gate," in *Advances in Cryptology - ASIACRYPT 2004*, ser. Lecture Notes in Computer Science, vol. 3329/2004. Springer Berlin / Heidelberg, 2004, pp. 119–136.

[5] D. Grigoriev and V. Shpilrain, "Zero-knowledge authentication schemes from actions on graphs, groups, or rings," 2008. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.118.7986

[6] B. N. Tran and T. D. Nguyen, "A graph isomorphism bases authentication protocol for access control in wlan," in *22nd International Conference on Advanced Information Networking and Applications*, 2008.

[7] J. V. del Campo, J. Hernandez-Serrano, and J. Pegueroles, "Profile-based searches on p2p social networks," in *The Ninth International Conference on Networks (ICN'10)*, April 11-16 2010.

[8] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub)graph isomorphism algorithm for matching large graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, p. 1367, 2004.

[9] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography," in *International World Wide Web Conference Comittee*, May 8-12 2007.

[10] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *Proceedings of the 24th International Conference on Data Engineering (ICDE'08)*, 2008.

Despliegue de políticas condicionadas para la protección de atributos en negociadores móviles

Carles Martínez-García*, Guillermo Navarro-Arribas†, Joaquin Garcia-Alfaro‡,§

* Universitat Autònoma de Barcelona, Edifici Q, 08193, Bellaterra
† Artificial Intelligence Research Institute, 08193, Bellaterra
‡ Universitat Oberta de Catalunya, 08018, Barcelona
§ Institut Telecom, Telecom Bretagne, 35576, Cesson-Sevigne, France

***Resumen*—En este trabajo, presentamos una propuesta para la realización de procesos automáticos de negociación en entornos móviles a través del despliegue de políticas condicionadas. Dichas políticas contienen un conjunto de atributos, condicionados por el nivel de confianza en el proceso de negociación, para proteger aquellas informaciones consideradas como sensibles o que potencialmente pudieran violar su privacidad. A medida que avance el proceso de negociación, los atributos servirán de mecanismo de control para concluir las distintas etapas que componen el proceso. Presentamos una visión práctica de nuestra estrategia orientada hacia su integración en aplicaciones pervasivas de comercio electrónico basadas en el uso de agentes móviles y políticas definidas según el estándar XACML.**

I. Introducción

Dadas las tendencias actuales en entornos de usuario de aplicaciones móviles y/o pervasivas, aparece con frecuencia un requisito básico: un proceso de negociación automático. El objetivo suele ser, por lo general, cubrir las necesidades de optimización de parámetros de dichos entornos. Algunos ejemplos apropiados podrían ser la optimización de parámetros en escenarios de tipo *roaming* (para conseguir los mejores parámetros de calidad de servicio) o *multi-homing* (para conseguir accesos de tipo *always best connected*). Por supuesto, es importante destacar también la necesidad de negociación con simples propósitos transaccionales (por ejemplo, búsqueda de una oferta, no necesariamente la más barata, en escenarios tradicionales de tipo proveedor-consumidor).

Todos estos procesos de negociación suelen requerir el intercambio de datos de carácter personal (tales como, preferencias de usuario, datos bancarios o histórico de negociaciones previas). Parte de esta información es generalmente vista por el usuario como sensible o privada. De hecho, este tipo de datos puede ser utilizado por los proveedores de manera inapropiada (por ejemplo, utilización ilícita de métodos de *profiling* para garantizar QoS). La necesidad de garantizar una protección adecuada de dichos datos se pone de manifiesto de manera aún más acentuada cuando el proceso de negociación se ejecuta, en nombre del usuario, por entidades *software* autónomas.

Estas entidades suelen ejecutar el proceso de negociación en plataformas distantes, diseminando registros sobre el usuario. Estos registros pueden ser utilizados por un adversario para violar su privacidad. Así, por ejemplo, trabajos como [7] han mostrado que tan sólo un mínimo de conocimiento sobre un individuo, cuya identidad se ha borrado de un proveedor de servicios, puede ser suficiente para identificarlo a partir de un conjunto de datos públicos. Es necesario, pues, introducir un mecanismo de protección adicional en estos entornos.

En este trabajo proponemos un mecanismo sencillo, a la vez que eficiente, para permitir la revelación progresiva de información de carácter personal en entornos pervasivos de negociación automática. Para ello, nos centramos en la realización del proceso por parte de agentes móviles. Estos presentan dos características importantes, movilidad efectiva de código (con todo lo que ello comporta), y capacidad de negociación en nombre del usuario. Así mismo, planteamos un acercamiento a la negociación basado en políticas, de manera similar a [9], [5], [1], donde agentes móviles se utilizan para automatizar la negociación entre clientes y proveedores de servicios. Nuestra propuesta se puede ver como una simplificación de sistemas de *trust negotiation* [8], [13], o como un soporte complementario a estas en entornos de computación móvil en general.

Organización del artículo — El resto del artículo se ha organizado de la siguiente manera: La sección II presenta algunos antecedentes en relación a nuestra propuesta. La sección III define de manera más concreta la motivación de nuestro trabajo y presenta la propuesta haciendo hincapié en su arquitectura y aportando notas de implementación. Finalmente, la sección IV concluye el artículo.

II. Antecedentes

Multitud de técnicas de negociación han sido propuestas y analizadas en el área de la matemática aplicada y de la teoría de juegos [4], [10]. La mayoría de estas técnicas tratan de utilizar modelos formales para definir e interpretar interacciones entre dos o más participantes, en forma de incentivos que conducirán finalmente el proceso de decisión de cada participante. Podemos encontrar, entre dichas soluciones, el estudio de estrategias que garanticen una decisión óptima, en términos económicos, a través de la detección de comportamientos preestablecidos. Estas técnicas han influenciado de manera decisiva la mayoría de soluciones basadas en la utilización de agentes y/o técnicas de computación cooperativa. De hecho,

estas soluciones han sido adaptadas con éxito hacia marcos de trabajo específicos en el área de negociación en aplicaciones de comercio electrónico [9] y redes IP móviles [5], [1]. La utilización de estas técnicas de negociación se prevé de vital importancia para las futuras aplicaciones de la tecnología ubicua, siendo de especial relevancia la necesidad de procesos de negociación que garanticen un intercambio mínimo de datos de carácter personal entre consumidores y proveedores de servicios electrónicos [12].

Cualquier proceso de negociación requiere la realización de una toma de decisiones. Estas decisiones son influenciadas, en gran medida, por las necesidades propias de cada una de las partes involucradas en el proceso. La capacidad de anticiparse a los deseos o motivaciones de un participante puede suponer una clara ventaja para sus oponentes. Por este motivo, la mayoría de sistemas de apoyo a la negociación tratan de anticiparse a los deseos/necesidades de sus oponentes mediante la incorporación de métodos que permitan modelar, y por lo tanto, anticipar, las decisiones de los participantes [11]. Asumamos, por ejemplo, aplicaciones de negociación electrónica de tipo *policy-driven*. Estas aplicaciones acostumbran a conducir el proceso de negociación a partir del intercambio de un conjunto de políticas en formato electrónico. Estas políticas permiten definir, a partir de lenguajes formales basados en lógica de primer orden, por ejemplo, el conjunto de declaraciones que será utilizado en el proceso de negociación. Multitud de lenguajes han sido propuestos en la literatura con el objetivo de formalizar este proceso [2]. Existen también en la literatura métodos de detección que permiten el análisis de dichas políticas para poder determinar, de entre un conjunto de posibles situaciones, aquellas que sean potencialmente más probables para conducir el proceso de negociación hacia un objetivo determinado [10], [11]. La parte más relevante a analizar acostumbra a ser el conjunto de atributos incluidos en la solicitud de ofertas, así como los resultados de la negociación. La figura 1 muestra un ejemplo, inspirado en la familia de protocolos propuestos en [9], donde podemos apreciar la inclusión de un conjunto de atributos que identificarán el objeto asociado al proceso de negociación, así como a las entidades involucradas en dicho proceso.

```
...
<Transaction name="SubmitProposal" ... >
  <Collaboration name="ReachAgreement">
    < InitiatingRole  name="Requester" id="... />
    <RespondingRole name="Responder" id="... />
    < Activity  name="RequesterNegotiation"
                binaryCollaboration="ConductNegotiation"
                fromAuthorizedRole="AgreementRequester"
                toAuthorizedRole="AgreementResponder">
      <Start toState="RequesterNegotiation" ... />
      <Transition  fromState="RequesterNegotiation"
                   toState="RequesterContract"
                   conditionGuard="Success"
                    ... />
      <Failure  fromState="RequesterNegotiation"
                conditionGuard="AnyFailure"
                 ... />
      <Success fromState="RequesterContract"
                conditionGuard="Success"
                 ... />
      <Failure  fromState="RequesterContract"
                conditionGuard="AnyFailure"
                 ... />
    </ Activity >
    <RequestOffer ... >
      <Attribute  name="currency" EUR />
      <Attribute  name="productNumber" 1234−5678 />
      <Attribute  name="productName" Notebook Computer />
      <Attribute  name="productDescription" Mobile ... />
    </RequestOffer>
</Transaction>
...
```

Figura 1. Solicitud de ofertas en una negociación de tipo *policy-driven*.

En el caso de negociaciones donde se requiera el intercambio de datos de carácter personal, ya sea para la identificación del objeto asociado al proceso de negociación, o para identificar a las entidades del proceso, es imprescindible garantizar un proceso de protección apropiado. La simple clasificación e identificación de los datos que requerirán dicha protección puede llegar a ser extremadamente compleja. Esto se pone aún más de relieve si tenemos en cuenta que incluso una dirección IP o un número de teléfono móvil asociado a las entidades u objetos del proceso de negociación pueden ser considerados como datos de carácter personal a proteger. En este sentido, el grupo de trabajo de la unión europea, encargado de regular la ley de protección de datos y vida privada de los ciudadanos, urge en su dictamen presentado en [3] la búsqueda de nuevas soluciones, más allá de la simple ofuscación de datos, para garantizar el derecho a la protección de datos de carácter personal de este tipo de aplicaciones. El objetivo de la propuesta que presentamos a continuación es precisamente iniciar el estudio de nuevas soluciones que puedan tratar esta problemática.

III. Información privada en negociadores móviles

El marco donde se centra nuestra propuesta se caracteriza por ser un entorno distribuido en el cual, por un lado, existen diferentes proveedores de servicios. Por otro lado, diferentes usuarios pretenden acceder a los distintos servicios desplegados. Como paso previo al acceso, se establece un proceso de negociación, local al proveedor, en el cual un agente móvil representa al usuario. Para ello, el agente negociador presenta una política de negociación que contiene información referente a éste. Esta información está contenida en una política XACML (*eXtensible Access Control Markup Language*), estándar de OASIS [6] que proporciona un lenguaje basado en XML muy flexible y expresivo, para especificar políticas de control de acceso, así como el protocolo de petición-respuesta asociado.

Dada la vulnerabilidad que supone que el código móvil se ejecute en una plataforma remota y, a priori, no confiable, hace que la información referente al usuario, contenida en la política de negociación, sea susceptible de ser comprometida. En términos de privacidad, esta vulnerabilidad desacredita a los agentes móviles a contener una política de negociación con información referente al usuario. No obstante, esta política se hace necesaria para completar el proceso de negociación. Cabe

destacar que cualquier registro de información residual en una plataforma podría ser utilizados por un adversario en beneficio propio. Surge, pues, la necesidad de controlar el filtrado de información de carácter personal. Es más, en ciertos casos en que el usuario no confíe lo suficiente en la plataforma del proveedor de servicios, se puede sacrificar el resultado final de la negociación por la preservación de cierto nivel de privacidad.

Con este propósito, el usuario percibe el riesgo de filtrado de información (invasión de privacidad) como un contexto de la misma negociación. Hay contextos donde el usuario, ya sea por su confianza en el proveedor de servicios o por su necesidad de obtener un resultado mejor en la negociación, puede estar dispuesto a revelar más información inicialmente considerada privada. Así pues, y dentro de cada contexto de privacidad concreto, proponemos que el agente móvil encargado de la negociación lleve una política ad-hoc a la etapa de negociación. Cada política que el agente negociador llevará consigo contiene información diferente según su grado de privacidad. De esta manera, se establece, mediante el contexto, el nivel de privacidad bajo el que debe operar el proceso de negociación. Aunque pueda parecer tedioso el desplegar una política diferente para cada contexto de privacidad, hay que tener en cuenta que la generación de políticas es completamente automática y, por lo general, transparente al usuario.

III-A. Arquitectura

El escenario en el que trabajamos está compuesto por distintas entidades [1]. En primer lugar, un agente móvil de tipo *User Negotiator* (UN), que es enviado por el usuario a la plataforma del proveedor. Su primera tarea es sondear las ofertas de los proveedores (fase de descubrimiento), y la segunda es la negociación del servicio. En segundo lugar, en el lado del proveedor, un agente de tipo *Access Negotiator* (AN) se encarga de negociar, con los UN que llegan, los términos de acceso al servicio. Por último, dada la vulnerabilidad que supone que un agente acarree información privada, surge la necesidad de aparición en el esquema de los agentes de tipo *User Overseer* (UO). Los agentes de tipo UO son los encargados de enviar agentes UN para el descubrimiento y la negociación de servicios. Estos agentes, además, son los encargados de gestionar la información que el UN contiene acerca del usuario durante cada fase de la negociación.

Como vemos en la figura 2, esta arquitectura no solo permite gestionar la información que es presentada al proveedor en cada fase de la negociación, sino también la cantidad de información que el agente negociador contiene sobre el usuario y, por ello, susceptible de ser comprometida. Se hace esencial un mecanismo entre UO y el UN que proporcione las funcionalidades necesarias para controlar el intercambio de información sensible. Dicho mecanismo se fundamenta de la siguiente manera:

- Definimos formalmente la confianza en el contexto de negociación como:
$$\sigma : [0, 1] \tag{1}$$
donde $\sigma = 0$ significa una total falta de confianza asociada a la negociación y $\sigma = 1$ significa el mayor nivel de confianza.
- Denotamos como $Attr_i$ al conjunto de atributos que representan al usuario. El usuario está interesado en obtener la mejor oferta de varios proveedores. El intercambio de datos $Attr_i$ durante la negociación entre UN y AN se condiciona por ciertas restricciones (por ejemplo, al resultado de alguna estrategia previa de negociación). De esta manera el UN debería intercambiar gradualmente $Attr_i$ preservando las preferencias del usuario.
- El revelado de información durante la negociación debe de ser gradual y consecuente al contexto de privacidad. Para ello, la criticidad de cada atributo se define como:
$$\mu(Attr_i) : [0, 1] \tag{2}$$
donde $\mu(Attr_i) = 0$ implica el menor grado de criticidad asociado al atributo $Attr_i$ y $\mu(Attr_i) = 1$ significa que el atributo posee el mayor nivel de criticidad. A su vez, el conjunto de atributos cuya criticidad es cero, esto es $\{Attr_i | \mu(Attr_i) = 0\}$, representa el conjunto de datos públicos que el usuario está dispuesto a intercambiar por defecto incluso antes de conocer el contexto de privacidad. Un ejemplo sobre la relación atributo-criticidad puede verse en el cuadro I.
- El usuario negocia con todos los proveedores de la misma manera. Por ello, inicialmente todos los proveedores se consideran no confiables.
- Dado un contexto de privacidad definido, el usuario revela cada uno de los atributos si y solo si:
$$\mu(Attr_i) \leq \sigma \tag{3}$$
Es decir, un atributo solo es revelado si su nivel de criticidad asociado es inferior o igual al umbral marcado por el nivel de confianza asociado al contexto de negociación.

En este caso, el UN es el sujeto que realizará la petición de la política que mantiene el UO. Asumimos que los agentes de tipo UN incluyen la funcionalidad necesaria para realizar el proceso de negociación [1] ya que pueden evaluar todas las peticiones respuestas y gestionar los contextos. Si aplicamos

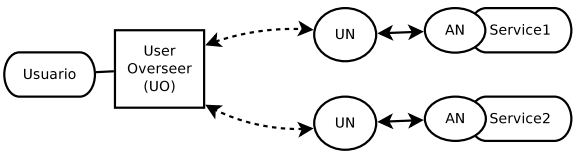


Figura 2. Arquitectura basada en [1].

Cuadro I
EJEMPLO DE LA RELACIÓN ATRIBUTO-CRITICIDAD.

$Attr_i$	$\mu(Attr_i)$
$attr_1$	0,8
$attr_2$	0,7
$attr_3$	0,3
$attr_4$	0

un refinamiento de políticas, se delega a los UNs la capacidad de decisión para controlar el intercambio privado de datos con los ANs. Una vez que los UNs son enviados a las plataformas de los proveedores, el usuario no puede —ni debe— interferir en la negociación.

III-B. Notas de implementación

El módulo UO, encargado de la generación dinámica de la política dependiendo del contexto de privacidad, presenta las siguientes funcionalidades:

- Recibir el nivel de confianza asociado a la negociación. El UN debe informar al UO del nivel de confianza asociado a la negociación con el proveedor de servicios.
- Representación de los atributos del usuario junto a su nivel de criticidad. El usuario debe, de forma sencilla, poder especificar los atributos que lo caracterizan así como su nivel de criticidad asociando a cada atributo un valor dentro del rango $[0, 1]$.
- Contener el patrón de la política de negociación. El UO debe contener el patrón de la política de negociación que le permitirá, una vez conocido el nivel de confianza asociado al contexto de negociación, generar la política adecuada.
- Presentar la lógica necesaria para la generación de la política de negociación de forma transparente al usuario. Esta lógica actúa sobre el patrón de la política, para generar de forma dinámica la política de negociación.

Para la generación automática de la política en XACML, existen varias herramientas que permiten la inclusión de código dentro de documentos de texto patrón. La ejecución de éste código redunda en un documento de texto alterado en el que el código ha sido substituido por el resultado de su ejecución. En nuestro mecanismo, usamos la herramienta de *templating* ERB [14]. De esta manera, se incluye, dentro de un fichero patrón que contiene la política de negociación expresada en XACML, la lógica necesaria, expresada en el lenguaje Ruby [15], para la generación de las líneas que relacionan los atributos con el usuario dependiendo del nivel de criticidad. La generación automática de código XACML es una idea análoga a la creación de servicios web dinámicos.

Previo al proceso de negociación, el usuario debe facilitar al UO tanto sus atributos como el nivel de criticidad asociado a cada uno de ellos. De la misma forma, el usuario debe facilitar el patrón de la política de negociación. Este patrón se compone de un documento XACML con cláusulas que contienen código en ruby. El usuario únicamente se debe preocupar por incluir una llamada a un método llamado *expand_attributes*. Método cuyo resultado de ejecución redunda en la impresión de aquellos atributos de usuario cuya criticidad sea menor o igual a la cota fijada por el contexto de privacidad. Cuando el UN establezca el contexto de privacidad, pedirá al UO que genere una política de negociación ad-hoc al contexto. Conociendo el contexto de privacidad, el patrón de la política de negociación y los atributos referentes al usuario, el UO es capaz de ejecutar la lógica incluida en el patrón para obtener la política de negociación que será enviada al UN. Una vez recibida la política, el UN continuará con el proceso de negociación.

```
...
<Subject>
  <%= expand_attributes %>
</Subject>
...
```

Figura 3. Patrón de la política de negociación.

La figura 3 muestra un ejemplo XACML simplificado del módulo que regula la revelación de atributos del UO, que generará las políticas de negociación del UN condicionadas por el contexto de privacidad. En dicha figura se observa que la inclusión de los atributos del usuario dentro de la política de negociación depende del contexto y se realiza a través de la llamada al método *expand_attributes*. La figura 4 muestra el resultado de la ejecución, dónde los atributos reflejados en el cuadro I se han incluido en la política de negociación bajo un contexto de privacidad $\sigma = 0{,}5$.

IV. Conclusiones

En este trabajo se ha presentado una solución de protección de la privacidad en procesos de negociación basada en un modelo de refinamiento de políticas condicionadas. El uso de un conjunto de transformaciones dinámicas de las políticas delegadas a un conjunto de agentes móviles se ha propuesto como mecanismo de implementación. Consideramos que nuestra propuesta podría ser también valida para otros escenarios, tales como procesos de integración de políticas de control de acceso, despliegue de configuraciones para sistemas de seguridad, o intercambio de alertas de detección en entornos de detección cooperativa. Una presentación más elaborada de nuestra propuesta así como su adaptación al resto de escenarios será tratada en un futuro informe.

Agradecimientos

Este trabajo está respaldado por el Departament d'Innovació, Universitats i Empresa (2009SGR1224), por la Universitat Autònoma de Barcelona (PIF 472-01-1/07) y por el Ministerio de Ciencia y Educación (proyectos E-AEGIS TSI2007-65406-C03-03, TSI2007-65406-C03-02, y CONSOLIDER-INGENIO CSD2007-00004 ARES). G. Navarro-Arribas disfruta de una beca Juan de la Cierva (JCI-2008-3162) del MICINN.

Referencias

[1] Benmammar, B., Jrad, Z., and Krief. F. QoS management in mobile IP networks using a terminal assistant. In *International Journal of Network Management*, 19(1):1–24, 2009.

[2] Bonatti, P., De Coi, J., Olmedilla, D., Sauro, L. Policy-Driven Negotiations and Explanations: Exploiting Logic-Programming for Trust Management, Privacy & Security. In *Logic Programming*, vol. 5366, LNCS, p. 779-784, 2008.

[3] Grupo de trabajo sobre protección de Datos del artículo 29. `http://ec.europa.eu/justice_home/fsj/privacy/docs/wpdocs/2007/wp136_en.pdf`

```
...
<Subject>
        <Attribute   AttributeId =" attribute "
         DataType="http://www.w3.org/2001/XMLSchema#string">
                <AttributeValue>attr3</AttributeValue>
        </Attribute>
        <Attribute   AttributeId =" attribute "
         DataType="http://www.w3.org/2001/XMLSchema#string">
                <AttributeValue>attr4</AttributeValue>
        </Attribute>
</Subject>
...
```

Figura 4. Política generada a través del patrón especificado en la figura 3.

[4] Karrass, C. L. Give and Take: The Complete Guide to Negotiating Strategies and Tactics. HarperCollins Publishers, New York, NY (1993).

[5] Krief, F. Self-aware management of IP networks with QoS guarantees. In *International Journal of Network Management*, 14(5), pp. 351-364, 2004.

[6] Moses, T. (Ed.). eXtensible Access Control Markup Language (XACML) Version 2.0. OASIS Standard, 1 Feb 2005.

[7] Narayanan, A., and Shmatikov, V. Robust De-anonymization of Large Sparse Datasets. In *IEEE Symposium on Security and Privacy 2008*, pp. 111-125, Oakland, California, USA (2008).

[8] Nejdl, W., Olmedilla, D., Winslett, M. PeerTrust: Automated Trust Negotiation for Peers on the Semantic Web. In *Workshop on Secure Data Management in a Connected World*, Canada, 2004.

[9] Rebstock, M., Thun, P., Tafreschi, O. A. Supporting Interactive Multi-Attribute Electronic Negotiations with ebXML. *Group Decision and Negotiation Journal*. vol. 12, p. 269–286, 2003.

[10] Stuhlmacher, A. F., Stevenson M. K. Using Policy Modeling to Describe the Negotiation Exchange. *Group Decision and Negotiation Journal*. vol. 6, p. 317–337, 1997.

[11] Vetschera, R. Preference Structures of Negotiators and Negotiation Outcomes. *Group Decision and Negotiation Journal*. vol. 15, p. 111–125, 2006.

[12] Yee, G. and Korba, L. Feature Interactions in Policy-Driven Privacy Management. *Seventh International Workshop on Feature Interactions in Telecommunications and Software Systems (FIW'03)*. 2003.

[13] Yu, T., Winslett, M., and Seamons, K. E. Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation. *ACM Transactions on Information and System Security (TISSEC)* vol. 6, no. 1, (Feb. 2003), 1-42.

[14] http://www.ruby-doc.org/stdlib/libdoc/erb/rdoc/

[15] http://www.ruby-lang.org/es/

Agregación de datos para autenticar información en VANETs

Jezabel M. Molina Gil, Pino Caballero Gil, Candelaria Hernández Goya, Cándido Caballero Gil
Departamento de EstadÍstica I.O. y Computación
Universidad de La Laguna
Email:{$jmmolina, , pcaballe, mchgoya, ccabgil$}@ull.es

***Abstract*—La comunicación inalámbrica entre vehículos conocida como Vehicular Ad hoc NETworking (VANET) permitirá proporcionar diferentes servicios y principalmente información a los conductores de manera que aumente la seguridad, eficiencia y confort en la conducción. En este tipo de redes los mensajes de advertencia repercutirán en las decisiones que tomen los conductores mientras circulan por la carretera. Por tanto, cualquier mensaje impreciso podría ocasionar pérdida de tiempo a los conductores, pérdida de dinero en cuanto a combustible, y en el peor de los casos accidentes. Por esta razón, un requisito indispensable para su uso es poder determinar si la información de tráfico vial que llega al conductor es significativa y de confianza. Validar este tipo de información sin que suponga un problema de sobrecarga y retardo en la red es casi imposible. En este trabajo proponemos una solución para validar la agregación de datos utilizando una comprobación aleatoria y probabilista de manera que permita descartar y descubrir intentos de ataques.**

I. Introducción

En la actualidad las VANETs tienen cada vez más importancia como motivo de estudio en numerosas investigaciones. Este tipo de redes permitirá en un futuro reducir e incluso evitar el número de muertes en las carreteras además de proporcionar información en tiempo real sobre el estado de las mismas. Por ejemplo permitirá a los conductores intercambiar información con sus vecinos advirtiendo sobre eventos potencialmente peligrosos como podría ser accidentes, obstáculos en la vía, etc. Otra utilidad para la que se han estudiado las VANETs es la posibilidad de encontrar plazas de aparcamientos libres en una determinada zona, evitar o reducir congestiones e incluso proporcionar información de las condiciones de tráfico en tiempo real.

Para que todas estas aplicaciones funcionen correctamente, es necesario asegurar que la información que circula en la red es fidedigna, por lo que será conveniente evitar o al menos disminuir el número de advertencias falsas en la misma. En la bibliografía actual podemos encontrar artículos que defienden la utilización de criptografía asimétrica en VANETs para, a través de la firma digital determinar de qué fuente llega la información y garantizar su integridad. Otros autores proponen la utilidad de criptografía simétrica para cifrar el contenido de la información proporcionando privacidad. Finalmente se propone el uso de seudónimos para proporcionar privacidad protegiendo la identidad de los usuarios. Sin embargo, todos estos mecanismos no nos protegen de un ataque sencillo y a la vez daino como es la generación de contenido falso. Un atacante podría inyectar información que no se corresponde con lo que está observando realmente. Por ejemplo, un conductor que tenga prisa por llegar a su destino, podría intentar diseminar información falsa acerca de una congestión en una determinada carretera para disminuir el número de vehículos que circulan por la misma.

Es cierto que gracias a la criptografía de clave pública sería posible determinar y sancionar al vehículo que presenta información falsa como verdadera. Sin embargo, el tiempo necesario para afrontar este problema por parte de las administraciones públicas hace que esta aproximación no sea útil dado que debe ser un mecanismo automático y en tiempo real. Para afrontar este aspecto proponemos utilizar la agregación de datos. Si bien es cierto que la agregación de datos se ha utilizado en muchos artículos como un mecanismo que permite disminuir el número de paquetes que circulan en la red, nosotros pensamos que además se puede utilizar para aumentar la fiabilidad de la información diseminada. En este artículo utilizamos la idea de cooperación y agregación de datos basadas en un esquema probabilista que proporciona seguridad a los datos de manera rápida y fiable.

La agregación de datos en VANETs ha sido introducida en algunos trabajos. En [1] Ibrahim presenta un protocolo para la retransmisión de información asumiendo que los vehículos forman clústeres. Detalles de velocidad e información se intercambian dentro del clústeres y cuando el clúster aumenta, los registros de información se agregan. Este mecanismo logra reducir la cantidad de datos transmitidos en un grupo de coches, pero no incluye mecanismos para combinar datos agregados. Otra propuesta de agregación se hace en [2] donde se presenta la agregación de varios mensajes que describen el mismo evento. Se propone también el uso de mensajes de revocación que permita a los vehículos denunciar mensajes falsos, un ejemplo sería el no detecta un peligro al entrar en una zona para la que se había recibido una advertencia. Este mecanismo presenta debilidades en cuanto a posibles ataques de adversarios los cuales pueden revocar mensajes que son reales. En [3] la solución propuesta se basa en el uso de un dispositivo tamper-proof y consiste en preguntarle a un vehículo agregador sobre un registro aleatorio agregado originalmente. La principal desventaja de este método es la dependencia del tamper-proof dado que un atacante fácilmente podría pasar por alto este servicio y componer agregados maliciosos. Finalmente [4] propone otro mecanismo para proporcionar seguridad mediante agregación en un esquema en el que las

calles se dividen en segmentos de tamao fijo correspondientes a la cobertura de las seales WiFi. Sin embargo, este criterio de agregación emplea una segmentación fija de la carretera. Se ha demostrado que este tipo de agregación no funciona bien con un gran número de vehículos y áreas más grandes, por ejemplo, en grandes atascos que abarquen kilómetros.

Este artículo se organiza como sigue: en la sección II se presentan conceptos básico sobre la agregación de datos y los modelos de adversarios. En la sección III presentamos la propuesta para generar los paquetes de agregación. En la sección IV discutimos acerca de los parámetros que se deben tener en cuenta a la hora de generar los paquetes de agregación. El mecanismo que permite determinar la autenticidad del mensaje se presenta en V. En VI y VII, analizamos el esquema y la fortaleza del mismo frente a los posibles ataques y finalmente en VIII presentamos las conclusiones.

II. Conceptos Básicos

Durante el estudio de las VANETs hemos realizado diversas propuestas para fomentar la cooperación y aumentar la seguridad en este tipo de redes con autenticación, gestión de claves y uso de seudónimos. Sin embargo, no encontrábamos herramientas que nos permitieran asegurar que la información que se generaba era cierta. Un planteamiento inicial y bastante lógico para hacer frente a esta necesidad era proporcionar a los nodos de un mecanismo que permitiera verificar la veracidad del contenido del paquete en el receptor. El modo de funcionamiento consistiría en proporcionar a los vehículos un almacén de información de modo que al recibir un paquete acerca de un peligro en la carretera, cada vehículo sería capaz de determinar la carretera y sentido de circulación donde se encuentra el mismo, el tipo de peligro y el nodo origen que generó este paquete. Una vez almacenada la información, el vehículo receptor debe compararla con otros paquetes que haya recibido conteniendo la misma información en la misma ubicación pero proporcionada por vehículos diferentes. En caso de no tener anteriores registros alertando del mismo peligro, el mecanismo de verificación tendrá dos opciones:

- Alertar al conductor del peligro y arriesgarse a que éste no sea cierto.
- No alertar al conductor y esperar a poder contrastar los datos pudiendo ocasionar un accidente.

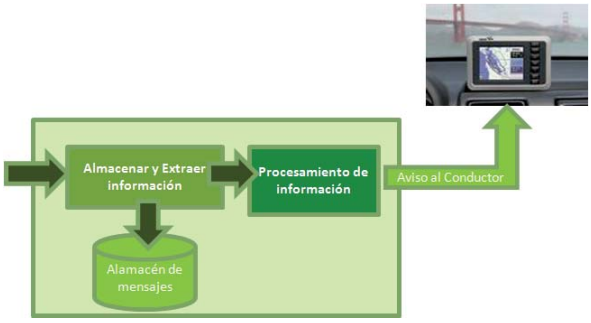


Fig. 1. Planteamiento Inicial de la Agregación

Analizando detenidamente estas alternativas vemos que en el primer caso, la información podría afectar en la decisión del conductor y podría traducirse en gasto de tiempo y/o dinero del usuario así como en un aumento de la desconfianza en el resto de mensajes que lleguen de la red. En el segundo caso, se producirá un retardo considerable debido a la espera de la llegada de un número suficiente de paquetes con el mismo contenido firmados por diferentes orígenes. Por consiguiente, el mecanismo debe ajustar el valor del número de paquetes a agregar de manera que el tiempo sea suficientemente corto como para advertir al conductor acerca del problema, y suficientemente grande como para poder asegurar que el contenido de la información es real habiendo sido contrastado por un número suficientes de vehículos.

Aparte de lo mencionado anteriormente, este mecanismo requerirá que el vehículo cuente con un importante espacio de almacenamiento así como con un mecanismo rápido para comparar diferentes registros. Esto supondría otro retardo que se le sumaría a la espera de los n paquetes a agregar. Es por esto por lo que una simple agregación de datos en el receptor no es suficiente para afrontar el problema.

Consideramos que un adversario es cualquier entidad que puede controlar varios vehículos en una cierta área de la red. El vehículo atacante puede participar activamente en la comunicación, es decir, puede enviar y recibir cualquier mensaje de agregación dentro de la red. Por una parte, centrándonos sólo en el proceso de agregación podemos identificar los siguientes ataques:

- ***Generar un mensaje individual de información falsa.*** Un atacante puede generar un mensaje en una vía que no se corresponde con la información real de su entorno. Por ejemplo, decir que circula a 20 km/h cuando en realidad circula a 80 km/h, con el objetivo de simular un atasco en dicha vía.
- ***Descartar un mensaje de agregación.*** Un atacante podría suprimir un mensaje agregado y como resultado no permitir el buen funcionamiento de la red.
- ***Generar un mensaje de agregación falso.*** Un atacante podría crear un mensaje agregado con información arbitraria e inyectarlo en la red como verdadero.
- ***Provocar repudio de un mensaje de agregación verdadero.*** Un atacante podría alterar alguno de los campos que permiten comprobar la veracidad de la información, con el fin de que los nodos de la red lo tomen como falso.

III. Modelo Propuesto

Debido a las características específicas de las redes vehiculares, la protección de la agregación de datos no es trivial. La alta movilidad de las redes hace que la topología cambie frecuentemente. Por lo tanto, los mecanismos de seguridad en este entorno no deberían asumir la existencia de estructuras estables. Pretendemos adaptar mecanismo de seguridad de [4], donde se utiliza una combinación de firmas junto con la idea de grupos, a una red donde no necesariamente deban formarse grupos explicitamente. Además se debe tener en cuenta que en la etapa de inicialización de estas redes no todos los vehículos

contarán con un dispositivo que les permita participar en la red como nodo y por lo tanto el esquema propuesto debe adaptarse a los distintos tamaos que presentará la red durante su vida. Este mecanismo permite la agregación de cualquier tipo de información, ya sea sobre incidentes en la carretera o con información relacionada con una conducción más confortable, etc. Es un mecanismo genérico que sirve tanto para redes autónomas como para aquellas en las que se requiere de una autoridad certificadora [5] permitiendo cualquier tipo de agregación que pueda hacerse en este tipo de redes. Este mecanismo de seguridad permite detectar ataques y mitigar sus efectos.

En nuestro esquema de agregación de datos, tenemos en cuenta diferentes aspectos de funcionamiento de los nodos: uno será aquel en el que mientras los vehículos circulan se encuentran con el obstáculo en la carretera y generan automáticamente mensajes de advertencia de peligro, otro será el de la recepción del paquete anterior confirmando que existe un peligro, y finalmente estarán los vehículos que reciban un paquete con la información y su respectiva confirmación. En el esquema básico cada vehículo retransmitía un mensaje de advertencia firmado, lo que suponía una considerable sobrecarga de la red, además del retraso resultante de la verificación y comparación de los datos procedentes de diferentes orígenes en el receptor. En este nuevo esquema proponemos combinar las firmas generadas por diferentes vehículos que avisen de un mismo problema. De este modo, combinamos las firmas y las agrupamos en un único mensaje, obteniendo como resultado un uso más eficiente de la comunicación inalámbrica. Sin embargo también este método presenta varios inconvenientes. Por una parte, al hacer una combinación de firmas, el tamao del paquete irá creciendo a medida que aumente el número de vehículos que confirmen la información por lo que volvemos a sobrecargar el canal. Por otro lado, el hecho de que la información esté firmada no significa que sea correcta. El receptor deberá en dicho esquema comprobar las firmas, lo que significa un retardo en la comprobación, que igualaría o incluso superaría el tiempo empleado por el modelo básico. Para solucionar este problema, proponemos aquí un tamao máximo de firmas que pueda contener el paquete de manera que no pueda crecer de manera infinita, y una granularidad basada en la idea de [6] donde se definen rangos dentro de los que podemos y debemos encontrar información. Finalmente, para solucionar el retardo ocasionado por la comprobación de firmas proponemos un esquema probabilista para comprobar algunas firmas. Todos estos mecanismos de seguridad se detallan a continuación.

IV. Tamao y granularidad

Tal cómo se comentó en el apartado anterior, el tamao del paquete se debe restringir a un máximo T que no sature el canal y que permita transmitir el paquete de manera rápida. En este caso el tamao deberá ser lo suficientemente grande como para tener suficientes testimonios de un mismo peligro sin exceder el máximo soportado por el canal inalámbrico. Teniendo esto en cuenta podemos definir ciertos criterios a

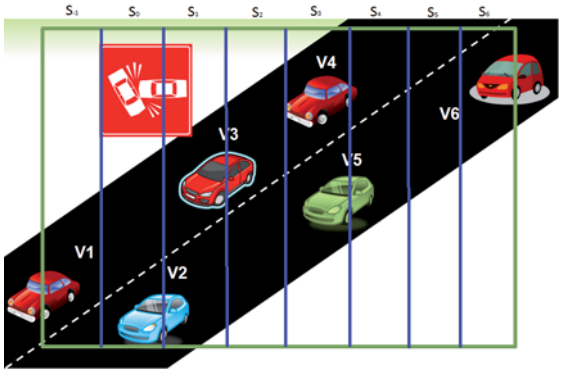


Fig. 2. Área de Peligro

la hora de introducir las firmas en un paquete de agregación. Por una parte, proponemos insertar en la primera y segunda posición del paquete los extremos que delimitan la zona de los datos detectados a agregar. Considerando que los datos agregados representarán un área donde los vehículos comparten valores comunes, se podrá obtener un área de los datos a agregar asociada a la localización del evento que se quiere difundir. De este modo si un vehículo es capaz de presentar información firmada validada sobre los bordes que delimitan un accidente en una agregación, esta información se puede tomar como válida. En la figura 2 los vehículos V1 y el V6 delimitan el área de peligro en un incidente.

Especialmente si el área agregada es grande, aadir valores de los bordes solo indicará que el dato agregado es válido, pero un adversario podría proporcionar valores para los bordes y no para el interior del área y de este modo mentir sobre la existencia un atasco. Se necesitarán más firmas que se correspondan con el interior del área. Para esto, además de delimitar el tamao del paquete a T firmas, se propone utilizar una granularidad S que consistirá en dividir las posiciones delante y detrás del incidente en regiones o celdas actuando del siguiente modo. Dependiendo del tipo de carretera el parámetro de granularidad S será mayor o menor teniendo en cuenta que cuanto más pequea sea la granularidad, mayor será la seguridad. El objetivo es seleccionar las firmas de manera que estén igualmente distribuidas en toda el área agregada. Así, antes de agregar una firma deberá estar a distancia S de otras dos como mínimo. En caso contrario no se introducirá o bien se cambiará la existente por ésta. Con esto se pretende tener información sobre un mismo incidente actualizada, distribuida y más fiable. A la hora de generar un paquete de agregación, un nodo deberá determinar la granularidad propuesta para el mismo. Esta granularidad dependerá del tipo de vía, siendo más grande en autopistas y más pequea para carreteras convencionales. Una vez definida la granularidad, las firmas empezarán a agregarse. Si un nodo es extremo lo indicará en la primera y segunda posición del paquete, y a continuación se irán aadiendo las diferentes firmas en los puntos de granularidad permitiéndose cierta variación. La estructura del paquete se define como sigue. El nodo que genera el paquete, aadirá la granularidad S así como una

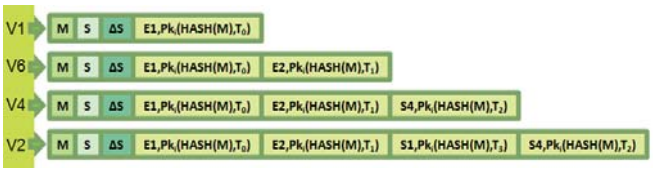


Fig. 3. Generación de paquete agregado

posible variación de la misma ΔS y reenviará el paquete, esto se corresponde al paquete generado por V1 en la figura 3. Cada nodo que reciba la información y detecte el mismo problema firmará el contenido y lo colocará en función del rango de granularidad al que pertenezca, en la figura 2 V1 y V6 eran los extremos por lo que sus firmas ocupan la primera y segunda posición. Posteriormente V4 agrega su firma y finalmente V2 que lo hace en la tercera posición dado que según la granularidad es la posicion que le corresponde.

Los nodos que son capaces de detectar un peligro no comprobarán las firmas contenidas en el paquete, sino que simplemente firmarán indicando que están de acuerdo con la información e introducirán la posición S_i en la que se encuentran con el fin de aligerar el proceso de creación de los paquetes de agregación. En cada agregación los nodos deberán primero ver la información, y si están de acuerdo con la misma, el primer paso será comprobar si son extremos o no. Nótese que los segmentos *S* se calculan respecto a la posición donde se generó el mensaje *M* y no respecto a los extremos, por lo que será fija. Si el nodo actual resulta ser extremo, cambiará el primer o segundo campo. Si el nodo no es extremo buscará la posición S_i a la que corresponde y firmará. Si resultara que la posición ya está ocupada, sólo se actualizará la información.

```
//Se reciben el mensaje M con las firmas F1...Fn
int Programa Principal
{     //Se crea un vector de tamaño igual al número de firmas a comprobar
      bool P[c];
      //Se crea un vector con tantos hilos como firmas hayan
      hilo H[n];

      for (cada hilo i que le toque comprobar ){
            /*Se elige si el hilo i comprueba con probabilidad p o
                                    no comprueba con probabilidad 1-p*/
            if (H[i]=1 )
                  P[j]=H[i](CompruebaFirma(F,M));
                  j++
            }
      }

      if(Todo P= verdadero){
            return mensaje fiable
      }
      else
      {
            if (Todo P = falso)
                  return mensaje no fiable
            else
                  return comprobar reputación
      }

/*Función que devuelve verdadero si la firma se corresponde con el testo
                                    y falso si no se corresponde*/
bool function CompruebaFirma(Firma Fi,Texto M )
{
      H[i] verifica Fi
      if Fi es valid then
            return true;
      else
            return false;
      end if
}
```

Fig. 4. Algoritmo de Verificación Probabilista

V. Verificación Probabilista

La verificación probabilista sólo se aplicará en aquellos vehículos que no son capaces de comprobar la información que les llega, es decir, cuando reciben un mensaje de advertencia en un punto que no es alcanzado por la cobertura de su antena. En este caso, el vehículo antes de tomar la información como cierta deberá comprobar que está firmada por diferentes vehículos. Como ya se ha comentado, es ineficiente comprobar todas las firmas contenidas en un paquete, pero si será necesario verificar la información antes de darla por válida y enviársela al conductor. Para solucionarlo nosotros proponemos verificar sólo un pequeo número de firmas. En esta sección, introducimos un esquema de autenticación que puede asegurar que el mensaje es auténtico sin verificar todas las firmas del mensaje recibido basándonos en COMET [7].

El algoritmo que se ejecuta en un vehículo se detalla en la figura 4 donde se utilizan hilos dado que son procesos más ligeros que permiten la ejecución concurrente. En el algoritmo, H_i, H_j y H_k son tres hilos donde i,j,$k = 1, 2, \ldots n$, con $i \neq j \neq k$. Cuando un vehículo recibe un mensaje se lanzan tantos hilos como firmas contenga el mensaje. Nótese que antes de este proceso se habrá comprobado si contiene suficientes firmas como para determinar que el mensaje se ha contrastado con un número significativo de vehículos. Cada hilo H_i determina si verifica la firma correspondiente con una probabilidad de verificación p. Si H_i realiza la verificación de la firma y es correcta devolverá un 1 indicando la veracidad de la firma y en caso contrario devolverá un 0. El resultado de todos aquellos hilos que han tenido que comprobar el mensaje se introducirá en una estructura P que esperará a que todos los hilos terminen su comprobación. Si todos los campos de *P* son verdaderos es que todas las firmas verificadas eran correctas por lo que el mensaje es válido. Por otro lado, si algunas de las comprobaciones resultaran incorrectas, podría significar que el mensaje es falso. En este punto, planteamos la posibilidad de utilizar la información de reputación almacenada por los vehículos y el número de respuestas de mensaje no fiable procedentes de diferentes vehículos que tenemos. Si hay una mayoría que indica que el mensaje es falso se tomará como falso y si ocurriera el caso contrario se tomaría como verdadero. Si existiese un empate o una cantidad dudosa, se comprobaría la reputación de los nodos, fiándose de aquellos que tienen buena reputación en cuanto a la cooperación en la red.

VI. Análisis de la solución

A continuación se analiza la solución propuesta. Para garantizar que un mensaje M_i es fiable, al menos una de las firmas del mensaje debe ser verificada. La probabilidad de que exista al menos una verificación debe ser un valor muy cercano a 1. Sin embargo, desde el punto de vista de la agregación de datos, una sola comprobación no es suficiente. Supongamos que se recibe un mensaje M_i con n firmas falsas y una verdadera y se trata de un mensaje falso. Si un hilo H_i verifica la firma verdadera, podría asegurar que este mensaje es verdadero. Por tanto, deben existir al menos dos comprobaciones de firmas para verificar un mensaje. Según las coordenadas del mensaje, a menos que se haya generado en un extremo, existirán firmas pertenecientes a vehículos que se encontraban por delante del mensaje generado y firmas por detrás. En caso contrario se dividirán por la mitad. Sea *n* el número total de firmas que contiene un paquete *Q*, dividimos el número de firmas en dos, *i* es el número de firmas de los vehículos que se encontraban delante de donde se generó *Q* (o la primera mitad) y *n-i* el número de firmas detrás de *Q* (o de la segunda mitad). Sea A_i el suceso de que hay *i* firmas que se generaron delante de *Q* y *n-i* detrás. Sea *B* el evento de que al menos se comprobarán dos de las firmas del mensaje, una de las cuales estará entre las que se generaron delante de *Q* y la otra detrás, esto evita comprobaciones de firmas que se generaron en una misma zona. Entonces la probabilidad $Pr\{B\}$ puede ser representada como una función de *n* y *p* como:

$$Pr\{B\} = \sum_{i=0}^{n} Pr\{B|A_i\} \cdot Pr\{A_i\} \tag{1}$$

$$= 1 + (1-p)^n - 2 \cdot (1 - \tfrac{p}{2})^n$$

Donde $Pr\{B|A_i\} = (1-(1-p)^i) - (1-(1-p)^{n-i}))$, y $(1-p)^i$ es la probabilidad de que ninguna de las *i* firmas por delante de donde se generó *Q* serán verificadas, $1-(1-p)^i$ es la probabilidad de que al menos una firma de las generadas será verificada y $1-(1-p)^{n-i}$ la probabilidad de que al menos una de las firmas por detrás de donde se generó *Q* será verificada;

$$Pr\{A_i\} = \binom{n}{i} \cdot (1/2)^i \cdot (1-1/2)^{n-i}$$

porque la posición donde se genera cada firma es independiente del número de firmas que se generan delante y detrás de donde se generó *Q*. Puesto que la posición de cada firma a verificar es independiente y el número de firmas delante y detrás sigue una distribución binomial con parámetro n y . El objetivo es hacer $Pr\{B\}$ tan cercano a 1 como sea posible.

En la figura 5 se muestra la relación entre $Pr\{B\}$, p y *n*, donde se puede ver como $Pr\{B\}$ aumenta cuando p y *n* aumentan. $Pr\{B\}$ se aproxima rápidamente a 1 cuando p es un valor pequeo. Además podemos concluir que cuando $Pr\{B\}$ es fija, p es inversamente proporcional a *n*. Nuestro objetivo es cambiar *p* de manera que $Pr\{B\}$ se aproxime a 1 tanto como sea posible. Por otro lado, cuando $Pr\{B\}$ se haya aproximado lo suficientemente a 1, intentamos hacer *p* lo más pequea posible porque un valor pequeo de *p* implica que un vehículo pueda potencialmente disminuir el procesamiento.

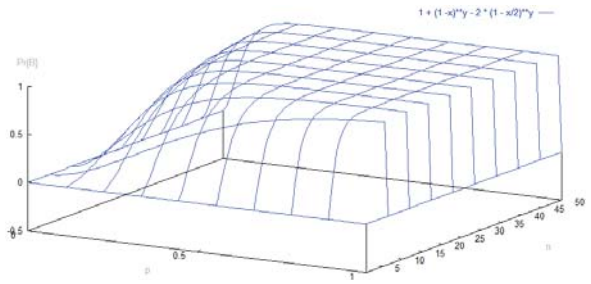


Fig. 5. Probabilidad de Éxito

A la hora de seleccionar el número de firmas que contendrá el paquete hay que tener en cuenta el tamao máximo que puede ser utilizado para este tipo de redes así como el número de firmas que se debe utilizar para asegurar la información. Para el primer caso tenemos tamaos de paquetes desde 256 bytes hasta 1500 bytes. Como en este tipo de redes se puede llegar a generar una gran cantidad de paquetes sería conveniente no utilizar el valor máximo dado que con un par de paquetes se saturaría el canal. Si tenemos en cuenta la función resumen que se va a utilizar para las firmas así como dejar unos 100 bytes para el contenido del mensaje, estaríamos hablando de 156 bytes en el peor caso y de 1400 en el mejor. Utilizando como función resumen SHA-1 que genera un resultado de 160 bits o lo que es lo mismo 20 bytes, generaríamos 7 firmas como máximo en el peor caso y 70 firmas en el mejor.

Para que cada vehículo elija un valor de *p* apropiado para los diferentes valores de *n* posibles, utilizamos el parámetro $k = n \cdot p$ para representar la proporción inversa entre *p* y *n*. Así *k* representa el promedio de firmas que un vehículo verifica porque *n* es el total de firmas contenida en el paquete y *p* es la probabilidad de verificación. Si encontramos un valor de *k* adecuado, entonces el valor de *p* correspondiente puede ser determinado fácilmente. Basándonos en (1), podemos obtener un relación entre $Pr\{B\}$ y *n* en términos de diferentes *k*, de forma que el valor de *p* puede ser determinado. Teniendo en cuenta que la probabilidad *p* tendrá como valor máximo 1 y habíamos dicho que como mínimo *n* sería 7, en la figura 6 vemos que con *k=7*, $Pr\{B\}$ es bastante cercano a 1 pero no lo suficiente. Sin embargo con k=10 la probabilidad es mucho más cercana a 1 cuando el paquete tiene 9 firmas o más. Por lo tanto, podemos fijar el valor de *k* a un valor constante, por ejemplo 10, y calcular *p* como *k/n*, que en este caso sería *10/n*, y de ahí modificar p según el valor de n. Por ejemplo, un vehículo que recibe un mensaje con 20 firmas, verificará cada firma con una probabilidad de 10/20. En caso de que el paquete contenga menos de 10 firmas, la probabilidad *p* para cada firma será del 100%

Teniendo en cuenta el número de firmas mínimo que puede contener un paquete para maximizar la probabilidad $Pr\{B\}$ además de calcular las probabilidades y el número de firmas máximos que cabe en un paquete, podemos determinar qué función resumen podemos utilizar en este tipo de redes. Si

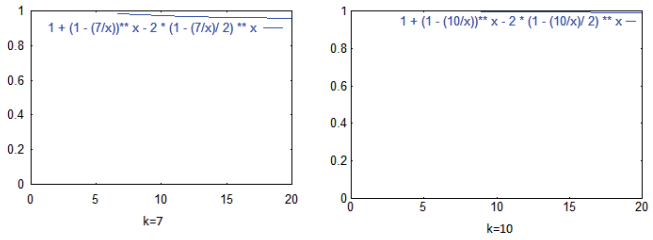


Fig. 6. Probabilidad con k=7 y k=10

Función Resumen	Tamaño del paquete	Tamaño para M	Tamaño para Firmas	Nº de firmas posibles
MD5	256	100	156	9
SHA-1				7
SHA-256				4
MD5	512		412	25
SHA-1				20
SHA-256				12
MD5	1024		924	57
SHA-1				46
SHA-256				28
MD5	1500		1400	87
SHA-1				70
SHA-256				43

Fig. 7. Funciones Resumen

nos fijamos en la tabla de la figura 7 y teniendo en cuenta que para una función hash utilizando MD5 se obtiene un resultado que ocupa 16 bytes, para SHA-1 20 bytes y para SHA-256 32 bytes y dejando un tamao de 100 bytes para datos en el paquete, vemos el número de firmas que podemos aadir en cada uno de los casos. Por ejemplo, con un paquete de 512 bytes y utilizando un total de 9 firmas con SHA-1, tenemos espacio suficiente para agregar las firmas necesarias. Incluso con este valor se permite utilizar una función resumen SHA-256 con paquetes de 512 y de este modo aumentar la seguridad de la función resumen.

VII. Análisis de Posibles Ataques

Como se presentó en la sección II se pueden intentar varios ataques con el fin de hacer que la red no funcione correctamente. El primer caso, donde se comenta la posibilidad de generar un mensaje con información falsa, queda descartado por la propia estructura de una agregación de datos. Los vehículos firmarán el mensaje si son capaces de detectar el problema que se especifica en el propio mensaje. El segundo ataque consistía en descartar un mensaje de agregación. Para solucionar este problema se han propuesto diferentes esquemas de cooperación [8]. Sin embargo el dao que pueda ocasionar a una agregación de datos no será demasiado grande dado que no sólo se generará un único paquete de agregación. El tercer ataque que consiste en generar un mensaje de agregación falso es posible llevarlo a cabo mediante la utilización de firmas de otros adversarios. Sin embargo, cuando un vehículo que no tiene acceso a dicha información recibe un mensaje de agregación tendrá que llevar a cabo dos comprobaciones. Por una parte, y con respecto a la agregación, deberán existir dos firmas correspondientes a los bordes del incidente, que pueden ser generadas por el propio adversario de manera correcta, pero aparte de eso, las firmas agregadas deberán cumplir con la granularidad y además se comprobará que las firmas se corresponden con el mensaje.

VIII. Conclusión

En este artículo se plantea la necesidad de afrontar un problema de seguridad existente en VANETs, que consiste en determinar si la información de tráfico vial que llega al conductor es significativa y de confianza. En concreto, proponemos un esquema para generar los paquetes de agregación de forma que sean seguros y difíciles de suplantar por un adversario. Para ello combinamos diferentes ideas en un nuevo esquema de agregación de datos de manera que aquellos vehículos que estén de acuerdo con la información generada firmen el paquete. Para evitar que el paquete crezca indefinidamente, las firmas se generarán siguiendo una granularidad definida según el tipo de vía y haciendo imposible por parte de un atacante la modificación de la misma. A su vez se generan dos firmas que delimitan la región. Si más de un vehículo coincidiera en granularidad se propone la actualización y reemplazo de firmas en una misma granularidad para mantener actualizada la información. Por otra parte, cuando el paquete agregado llegue a un vehículo, éste podrá verificar su información comprobando la firma. Para evitar el retardo que supondría tener que comprobar todas las firmas se propone un esquema probabilista de manera que se elige comprobar alguna de las firmas siendo el número de firmas a comprobar un punto de equilibrio para asegurar que la información es cierta, y no provoca retardos en la obtención de la información.

Agradecimientos

Investigación financiada por el Ministerio de Educación y Ciencia y la fundación Europea FEDER bajo el proyecto TIN2008-02236/TSI, y la Agencia Canaria de Investigación, Innovación y Sociedad de la Información bajo el proyecto PI2007/005.

References

[1] K. Ibrahim, M.C. Weigle, “Accurate data aggregation for VANETs“,en *P*roceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks, VANET ’07, 2007

[2] S. Eichler, C. Merkle, and M. Strassberger, “Data aggregation system for distributing inter-vehicle warning messages,“en *3*1st IEEE Conf. on Local Computer Networks, pp. 543–544. IEEE Computer Society, November 2006.

[3] F. Picconi et al., “Probabilistic Validation of Aggregated Data in Vehicular Ad-Hoc Networks,“ en *P*roceedings 3rd Int’l. Workshop Vehicular Ad Hoc Networks, ACM Press, pp. 7685, 2006.

[4] M. Raya, A. Aziz, and J. Hubaux, “Efficient Secure Aggregation in VANETs,“ en *P*roceedings 3rd Int’l. Workshop. Vehicular Ad Hoc Networks, 2006, pp. 67-75.

[5] A. Viejo, F. Sebé, J. Domingo-Ferrer, “Aggregation of Trustworthy Announcement Messages in Vehicular Ad Hoc Networks“ en *I*EEE Vehicular Technology Conference, 2009.

[6] S. Dietzel, E. Schoch, B. Konings, M. Weber, and F. Kargl,“Resilient secure aggregation for vehicular networks“ en *I*EEE Network, 2010.

[7] C. Zhang, X. Lin, R. Lu, P.-H. Ho and X. Shen, “An Efficient Message Authentication Scheme for Vehicular Communications“ en *I*EEE Transactions on Vehicular Technology, 2008.

[8] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, “A Vision of Cooperation Tools for VANETs“ en *P*roceedings of the First International Workshop on Data Security and PrivAcy in wireless Networks, 2010.

Gestión de Grupos en VANETs: Descripción de Fases

C. Caballero-Gil, P. Caballero-Gil,
J. Molina-Gil, C. Hernández-Goya
Departamento de Estadística,
Investigación Operativa y Computación,
Universidad de la Laguna
Email: {ccabgil, pcaballe, jmmolina, mchgoya} @ull.es

A. Fúster-Sabater

Departamento de Tratamiento
de la lnformación y Codificación,
Instituto de Física Aplicada, (C.S.I.C.)
Email: amparo@iec.csic.es

***Abstract*—En este trabajo se analizan algunos problemas inherentes a las VANETs debidos la falta de infraestructura central y a que los usuarios están en constante movimiento, dificultando conexiones, enrutamiento y seguridad de las comunicaciones. Además, otro inconveniente que también debe ser resuelto es la escalabilidad ya que el número de vehículos que circulan por la carretera es cada día mayor. Ante situaciones de tráfico denso de vehículos que ocasionarían que las comunicaciones se degradasen, este trabajo propone el uso de grupos como solución al intercambio de información ya que utilizando grupos se puede fraccionar la VANET en pequeñas subVANETs que permiten por una parte no enviar la misma información por múltiples caminos, y por otra, usar criptografía de clave secreta. De esta forma se logra mejorar la eficiencia y la seguridad de las comunicaciones. En particular aquí se proporciona una definición completa de los procesos de creación y administración de grupos en VANETs, así como de la forma de tratar la información cuando los vehículos pertenecen a un grupo.**

I. Introduction

Una red ad-hoc vehicular o VANET (Vehicular Ad-hoc NETwork), como su propio nombre indica, es una red ad-hoc donde sus nodos se corresponden con vehículos. Concretamente se trata de una red de tipo MANET (Mobile Ad-Hoc NETwork), es decir, una red móvil ad-hoc formada espontáneamente por nodos en pleno movimiento.

La seguridad vial es un problema de vital importancia clasificado por la comunidad Europea como uno de los objetivos prioritarios. Existen múltiples situaciones en las que las comunicaciones entre vehículos ayudarían a prevenir accidentes y evitar atascos. Cada año se producen más accidentes de tráfico, debido en gran medida a que cada año existen más vehículos en la carretera. Proporcionar información al conductor acerca de los peligros existentes podría ayudar a prevenir y evitar riesgos. Sólo un 6% de los accidentes es inevitable y está fuera del alcance de las mejoras tecnológicas. El resto podría evitarse usando dispositivos que alertaran al conductor de los peligros de la carretera. Además, las VANETs ayudarían a evitar atascos, ahorrando así tiempo, dinero, contaminación del medio ambiente, reservas de petróleo y destrucción del paisaje por construcción de carreteras.

Existen varias características de las comunicaciones que deben ser protegidas en cualquier red inalámbrica: autenticidad, privacidad, anonimato, cooperación, bajo retardo, estabilidad de las comunicaciones, etc. En particular, en las VANETs los escenarios son muy cambiantes, desde carreteras comarcales sin apenas tráfico hasta ciudades o autopistas plagadas de vehículos donde el número de comunicaciones es casi infinito [8], [7], [1]. En este trabajo definimos esquemas para la utilización de grupos en VANETs que permiten a la vez optimizar considerablemente las comunicaciones en situaciones de tráfico denso, y definir claves secretas de grupo con las que se podrá usar criptografía de clave secreta para distribuir información de forma rápida y segura.

En cuanto a la utilización de grupos o clusters en VANETs, el trabajo [2] presenta un análisis teórico de un algoritmo de agrupamiento basado en la estabilidad direccional. En [4] se proponen clusters donde el líder es el nodo que se encuentra en medio del grupo y tiene el identificador más bajo. [3] describe clusters con 4 fases: NoDecidido, Miembro, Entrada y CabezadeCluster. En [10] se proponen clusters para maximizar el avance de la información retransmitida y evitar interferencias. [5] propone la firma de grupos para proteger la privacidad, siendo las Road-Side-Units (RSU) los distribuidores de claves. Estos documentos no definen en detalle cómo se producen los cambios en el grupo, lo que es nuestro principal objetivo.

En la sección II se mencionan algunas aplicaciones y tipos de enrutamiento que pueden usarse en estas redes. A continuación se proporciona una definición de grupo y una propuesta para el establecimiento de clave secreta compartida. En la sección IV se explican en detalle las distintas fases de grupos que se distinguen en este trabajo: Detección, Creación, Elección, Entrada y Final. La sección V describe cómo se realizan las comunicaciones en el grupo. Finalmente se detallan las conclusiones.

II. Aplicaciones y Enrutamiento

Las aplicaciones de las VANETs en seguridad vial puede clasificarse en tres categorías:

- incidentes dinámicos (infracción grave detectada, circulación en sentido contrario, vehículos de emergencia, etc.)
- incidentes estáticos (límite de velocidad, obras en carretera, etc)
- gestión de incidentes (accidente, atropello, etc.)

Otras aplicaciones más avanzadas son la notificación de servicios, los esquemas de asistencia cooperativa al conductor y los sistemas de monitorización. Por último, existen una serie de aplicaciones interesantes que aportan valor añadido a las VANETs, incluyendo por ejemplo el anuncio de zonas de interés, la comunicación interactiva entre las OBUs (On-Board-Units) situadas en los vehículos o el pago electrónico, entre otras. Una característica importante de este tipo de servicios y aplicaciones comerciales es que no deben interferir con las aplicaciones de seguridad vial ya que éstas son prioritarias, tal y como queda reflejado en el orden de ejecución descrito en el DSR (Dedicated Short Range) incluido en el borrador de estándar para VANETs IEEE802.11p WAVE (Wireless Access for Vehicular Environments). Diferentes tipos de encaminamiento para la entrega de información dentro de una VANET son necesarios según el tipo de comunicación y su aplicación. Los modos más relevantes son los siguientes:

- Unicast: Envío uno a uno, es decir asociación con un único emisor y un único receptor.
- Broadcast: Difusión de uno-a-muchos, donde se pretende que la información sea recibida por todos los posibles nodos receptores de la red.
- Multicast: Envío de uno-a-muchos, donde la información replicada y se envía a un conjunto específico de receptores.
- Geocast: Multidifusión especial limitada basada en las posiciones geográficas de los nodos.
- Anycast: Envío en el que sólo uno de los posibles receptores se elige para recibir la información.

III. Grupos y Clave Secreta

Llamamos aquí grupo a un conjunto de nodos (vehículos con OBU) que se encuentran cercanos entre sí, situados en una zona geográfica, y dentro de la cobertura de al menos uno de ellos. Definiremos la distancia entre nodos en función del número de saltos (hops) mínimos necesarios para la comunicación entre ellos. Para pertenecer al mismo grupo todos los nodos deben estar a distancia uno de al menos uno de ellos. Para hacer más simple y ligera la gestión del grupo, uno de esos nodos que recubren el grupo será elegido como líder del grupo. Por tanto, los grupos se basan en la proximidad geográfica y están acotados a un salto de distancia. El líder de grupo será el encargado de gestionar la información y las conexiones que se establecerán dentro y con su grupo, delegando trabajo a otros vehículos si fuese necesario.

La gestión de grupos debe cumplir dos requisitos: minimizar el consumo de recursos y en particular el intercambio de mensajes, y tener en cuenta la topología altamente dinámica de la red, requiriendo actualización de la pertenencia al grupo. Se exige un mínimo de vehículos para constituir un grupo. En particular, los vehículos formarán grupos según celdas dinámicas donde el líder es el vehículo con tecnología VANET que haya iniciado el grupo que tenga mayor número de vecinos en el momento en el que el anterior líder caiga por debajo del umbral definido para la formación de grupo. Esta definición espontánea de los grupos en general está en función de la velocidad media de la vía y la dirección en la que circulan los vehículos, de forma que los vehículos que circulen a una velocidad cercana a la media en la mayor parte de los casos no cambian de grupo durante su trayecto.

Los grupos sólo serán usados cuando las condiciones de la vía así lo requieran, por ejemplo, con tráfico denso, atascos o carreteras muy concurridas, donde la densidad de vehículos en una zona geográfica hace que el número de comunicaciones sea excesivamente grande provocando colisiones en la señal de la información enviada y degradando notablemente la calidad de las comunicaciones, así como para posibilitar el uso de criptografía de clave secreta en VANETs. En los casos donde el número de vehículos sea bajo y no exista saturación de las comunicaciones no se activarán las comunicaciones por grupo.

Si n denota el número de nodos, la propuesta basada en grupos implica una significativa reducción en el número de envíos broadcast. En particular, sin la utilización de grupos cada uno de los n nodos envían mediante broadcast cada dato a enviar de forma que el número total $n(n-1)$ de comunicaciones crece rápidamente debido a las colisiones entre paquetes. Sin embargo, utilizando el esquema de grupos propuesto, se generan 3 conexiones por grupo para cada dato enviado ya que se reducen las colisiones. La primera conexión va desde el vehículo que produce la información hacia el líder (a no ser que la produzca el mismo líder). A continuación el líder hace un envío a todos los vehículos del grupo. Por último, habrá otra conexión entre el líder y otro vehículo (el que se encuentre en la mejor posición para continuar la difusión de la información). Por tanto se generarán un total de $3*$ (número de grupos) envíos de información por cada paquete de datos en las zonas de alta densidad de tráfico.

La mayoría de las referencias bibliográficas proponen como solución criptográfica para las comunicaciones en VANETs el uso de una Infraestructura de Clave Pública, con certificados expedidos por una Autoridad Certificadora. Esta solución supone que a cada nodo se le asigna un par de claves pública/privada que se almacena en un dispositivo tamper-proof. La propuesta de gestión de las comunicaciones basada en grupos permite el uso de criptografía de clave secreta ya que posibilita a los miembros del grupo llegar a un acuerdo sobre una clave secreta compartida. La comunicación con clave secreta compartida y la cercanía de los miembros del grupo, que se comunican en modo promiscuo, permite a los nodos pertenecientes a un grupo controlar que tanto el líder como sus compañeros de grupo actúan correctamente. En particular en el esquema descrito a continuación los nodos que forman un grupo deben generar una clave secreta compartida. Para ello el esquema que proponemos se basa en la dificultad del problema de logaritmo discreto, consistente en describir el valor de S a partir del conocimiento de $g^s(mod\ p), g$ y p. Este problema es la base del conocido método de Diffie-Hellman para el intercambio secreto de claves entre dos usuarios, de forma que la propuesta de este trabajo puede verse como una generalización de dicho método para un conjunto de usuarios.

Por otra parte el esquema se basa en el uso de esquemas de compromiso de bits. Los nodos se comprometen frente al líder

con su aportación a la clave secreta compartida de forma que el líder no pueda cambiar dicha aportación, ni leerla. Asímismo, el uso del esquema de compromiso hace factible el intercambio de información pública para la generación por cada nodo del secreto compartido sin que el secreto de la clave compartida ni de las distintas aportaciones corran peligro.

La notación utilizada en el esquema es:

h : función hash.
p : número primo.
g : elemento generador de Z_p.
i : nodo.
L : líder de grupo.
S_i : entero aleatorio seleccionado por i entre 0 y p-2.
g^{S_i} : compromiso público de i con entero S_i.

1) Cada nodo i que acepte formar parte del nuevo grupo envía al líder L su compromiso $g^{Si}\,(mod\ p)$
2) El líder L hace broadcast del mensaje $\left\{h(g^{S_i}), g^{S_i S_L}\right\} \forall_{i \neq L}$
3) Cada nodo i que acepte formar parte del grupo puede construir la clave secreta compartida K ya que conociendo S_i, puede calcular el inverso de $S_i\,(mod\ p-1)$ y por el teorema de Euler g^{S_L}, de forma que $K = g^{S_L} * \Pi_{i \neq L} g^{S_i S_L} = g^{S_L\left(1+\sum_{i \neq L} S_i\right)}$

De esta forma la clave del grupo es generada mediante las aportaciones de los primeros miembros del grupo. Mientras el grupo exista, las nuevas incorporaciones de nodos se realizan mediante el envío cifrado de esta clave secreta con la clave pública de cada nuevo nodo.

IV. Fases de Grupos

Las VANETs son redes inalámbricas en las que puede haber un gran número de conexiones altamente volátiles entre vehículos. Por este motivo se hace necesario definir en detalle la forma en la que deben actuar los vehículos según la situación en la que se encuentren. En el presente trabajo se exponen diferentes fases en las que puede encontrarse un vehículo según la vía y la situación en la que el vehículo se encuentre. Concretamente se distinguen: Detección, Elección, Creación, Entrada y Salida del Grupo. El esquema global de la vida de la red propuesto en este artículo es como sigue. Inicialmente todos los nodos comienzan en la fase de Detección de Grupo. Tras ésta, pueden entrar en la fase de Creación o de Elección de Grupo, dependiendo de las circunstancias. Tras la fase de Creación de Grupo, el nodo sería el líder de grupo y entraría en la fase de Vida del Grupo. Tras la fase de Elección de Grupo, el nodo entraría en la fase de Pertenencia a Grupo. El esquema es cíclico dado que tras terminar las fases de Vida de Grupo o Pertenencia a Grupo el nodo comienza nuevamente en la fase de Detección de Grupo.

A. Detección de Grupo

Esta primera fase es en la que cada vehículo estará normalmente, es decir, cuando no se encuentra en una situación de tráfico denso. La figura 1 muestra cómo el vehículo comprobará cada cierto tiempo el número de vecinos, y el número de líderes entre ellos. En caso de existir al menos un vecino que sea líder de un grupo el nodo pasará a la fase de elección de grupo, y en caso contrario a la fase de creación de grupo. Esta fase no genera ningún tipo de tráfico de control dado que toda la información necesaria está contenida en los beacons que lanzan los usuarios para su descubrimiento.

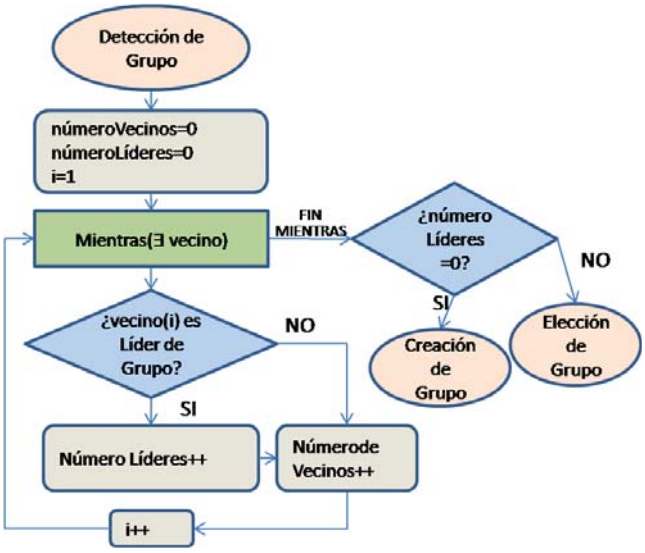


Fig. 1. Detección de Grupo

B. Elección de Grupo

En esta fase se supone que el vehículo ha encontrado entre sus vecinos al menos un nodo que es líder de algún grupo. En caso de que sólo haya un vecino que es líder de grupo, la elección es automática. En caso de que haya varios, el nodo tendrá que elegir a cuál de ellos unirse. En la figura 2 se puede ver cómo resuelve las distintas posibilidades que pueden darse. Para ello tendrá en cuenta tres valores para cada grupo j:

- La densidad $A\,(j)$ de vehículos.
- La calidad media de la señal $B\,(j)$ de los vehículos.
- Los segundos $C\,(j)$ que ha estado conectado al líder.

Todas estas variables tomarán valores que serán determinados a partir de pruebas y simulaciones donde se comprobará cuál de estas características es más importante. Una vez elegido el grupo al que se va a conectar, el vehículo envía una petición de entrada junto con su clave pública al líder del grupo el cuál después de autenticarlo, le enviará la clave secreta del grupo cifrada con la clave pública del vehículo receptor, momento a partir del cual dicho vehículo pasa a formar parte del grupo.

C. Creación de Grupo

En la fase de creación de grupo (figura 3) el vehículo no tiene cerca a ningún líder de grupo. Debe comprobar si entre sus vecinos existen al menos X que no pertenecen ya a un grupo, más un número variable Y que indica el número de vehículos que pueden apagarse, separarse o no unirse al nuevo grupo que se va a crear. En caso de que el número de vehículos

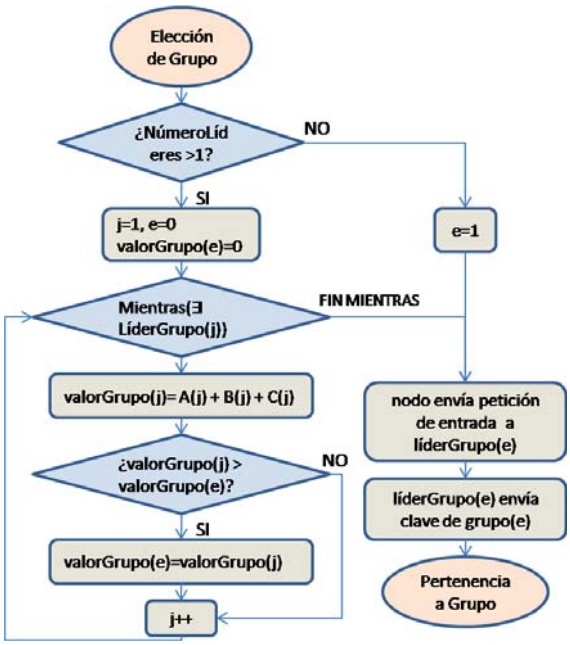


Fig. 2. Elección de Grupo

vecinos sin grupo sea inferior al mínimo requerido para esta formación de grupo, el vehículo espera un periodo $tiempo1$ y comienza de nuevo la fase de Detección de grupo.

Si el número de vecinos supera el umbral $X+Y$, el vehículo comienza un proceso de creación de nuevo grupo. Para ello realiza un multicast hacia todos los nodos vecinos a distancia 1 con petición de creación de grupo, los nodos que la reciben responden aceptando o rechazando la invitación. Si el número de vecinos que aceptan supera el umbral mínimo X, el nuevo líder de grupo enviará el mensaje que permita a dichos nodos reconstruir la clave secreta de grupo. En caso contrario, si el número de vehículos que aceptan la formación de grupo no supera el límite X, se incrementa el valor Y sumando la cantidad de vehículos que no aceptaron la invitación. En conclusión, en esta fase son necesarios un multicast de invitación a nuevo grupo, una respuesta unicast por parte de n usuarios y finalmente un broadcast para retransmitir el mensaje que permita a los nodos del grupo construir la clave secreta de grupo. En total $2n+1$ paquetes en caso de formación de grupo, y $n+2$ en caso de que el proceso falle.

La creación de grupo se inicia cuando el número de vecinos adecuados alcance un cierto umbral de tráfico denso, cuando el exceso de comunicaciones sin grupos degrada la red. Por tanto, los paquetes de gestión generados en esta fase no influyen negativamente en el funcionamiento de la red.

D. Final de Grupo

Una vez esté conformado el grupo, el líder debe validar cada cierto tiempo que el grupo sigue siendo válido, para en caso contrario, cambiar de líder o deshacer el grupo. La figura 4 muestra el proceso para la salida de un nodo del grupo al que pertenece. Cuando dicho nodo pierde el contacto con el líder del grupo durante un tiempo definido, el nodo deja de

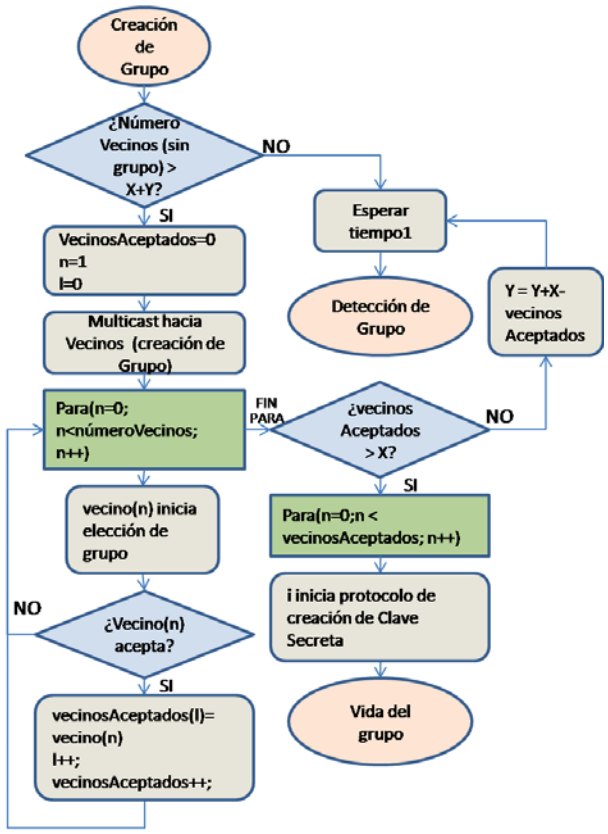


Fig. 3. Creación de Grupo

pertenecer al grupo al que pertenecía y comienza la fase de Detección de grupo en caso de que la densidad de nodos sea superior al umbral. La salida, al igual que el resto de fases de grupo se hacen de forma automática sin que el usuario note ninguna diferencia en el comportamiento del dispositivo.

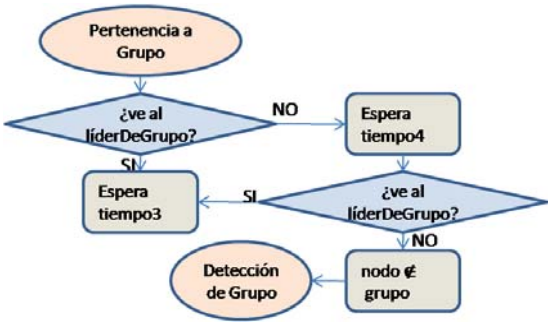


Fig. 4. Pertenencia a Grupo

La figura 5 muestra cómo el lider del grupo comprueba periódicamente que el grupo sigue siendo útil. En caso de que el tamaño del grupo caiga por debajo de cierto umbral, el líder comprueba si tiene un número de vecinos mayor o igual a D (umbral de tráfico denso) espera un $tiempo2$ en vez de iniciar un protocolo de final de grupo para no introducir tráfico de gestión cuando el vehículo se encuentra en una situación de tráfico denso. En caso de no estar en una situación de tráfico

denso, el actual líder comenzará un proceso de cambio de líder o final de grupo. Para ello preguntará a sus vecinos la densidad de su vecindario y tras esto sabrá si existen vecinos con densidad de vecindario suficiente (con número de vecinos pertenecientes al mismo grupo o sin grupo asignado mayor que X) y cuál de ellos tiene la mejor situación (mayor número de vecinos pertenecientes al grupo en su alcance de transmisión). Después enviará una señal multicast de cambio de líder a todos sus vecinos y el nuevo líder iniciará una fase de creación de grupo con los nodos sin grupo que están en su rango de transmisión. En caso de no existir ningún vecino que supere dicho umbral, el líder enviará mediante multicast la señal de final de grupo a todos sus vecinos.

Esta fase de grupo es la que más comunicaciones requiere pero el número de paquetes retrasmitido no influye negativamente en la red dado que sólo se lleva a cabo cuando el número de nodos en el rango de transmisión no alcanza el umbral D de alta densidad. En esta fase se genera un paquete multicast de petición de potencial de cada nodo (número de vecinos pertenecientes al grupo más vecinos no pertenecientes a ningún grupo). Responderán sólo los nodos cuyo potencial supere el umbral de grupo X. En el peor de los casos esto implica n unicast hacia el líder. Finalmente se requiere un multicast desde el líder hacia el resto de los vecinos con la señal que corresponda: cambio de líder o final de grupo. Si es una señal de cambio de líder, el nuevo líder inicia una petición de creación de grupo. Por tanto en esta fase se generan en el peor de los casos $n + 1$ (número de vecinos del líder) $+m + 1$ (número de nodos no pertenecientes a ningún grupo en el alcance del nuevo líder). Obsérvese que el número de paquetes generados es mayor que si se disolviera el grupo y se volviera a crear, pero sin embargo, el coste computacional de construir una nueva clave secreta compartida por los vecinos es alto por lo que evitar tener que generar dicha clave prima sobre el número de paquetes generados.

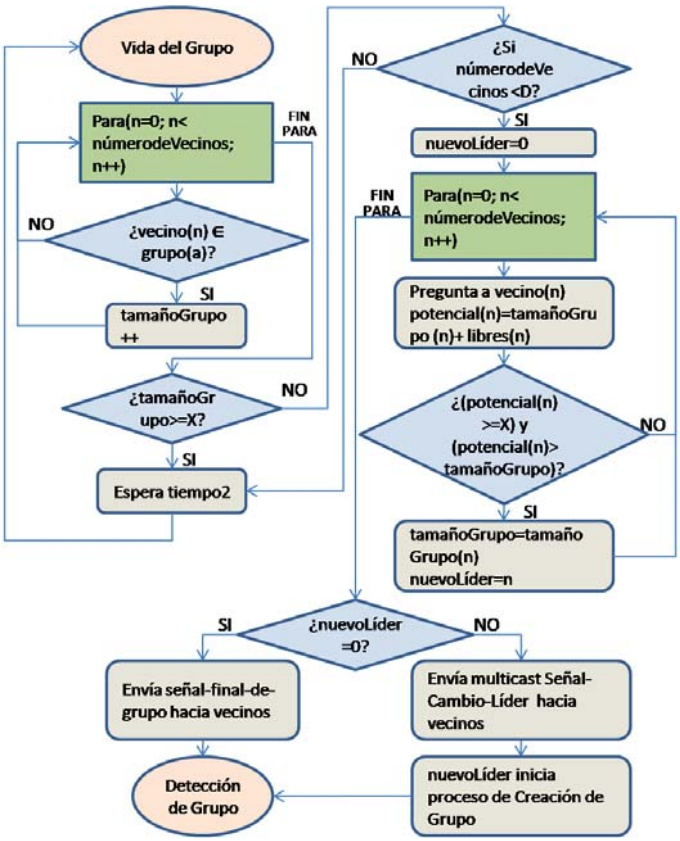


Fig. 5. Vida del Grupo

V. Tratamiento de mensajes en el Grupo

Tal y como se ha destacado, el número de comunicaciones que se producen en las VANETs en momentos de tráfico denso es inmenso. Por este motivo una buena gestión de las comunicaciones se hace necesaria en dichos casos. La utilización de grupos formados antes de llegar a una situación de tráfico denso permite gestionar las comunicaciones de forma eficiente enviando el menor número de paquetes posible sin perder ningún tipo de información útil.

En la figura 6 se explican los pasos que un vehículo conectado a un grupo debe seguir para tratar una señal de entrada. En primer lugar debe comprobar si el paquete que le llega es parte de una secuencia que está reenviando. Si es así, debe seguir con la secuencia y reenviar el paquete hacia el nodo destino, y cuando se completa la secuencia se termina el proceso de recepción. En caso de que el paquete no sea parte de una secuencia, se distingue si el vehículo es el líder del grupo o no. En este último caso, si el dato no va dirigido hacia el vehículo, comprueba si el dato lo envía el líder del grupo. El líder de grupo envía dos tipos de paquetes a nodos del grupo no destinatarios:, conexión a internet para secuencia de información, o paquete de información pública que debe ser reenviado hacia otras partes de la red.

El líder es el encargado de tratar los datos que llegan al grupo. Esto no significa que sea el líder el que comprueba que todos los datos son correctos. En su lugar, el líder gestiona la información que llega al grupo y reparte la comprobación de la información entre el resto de nodos del grupo indicando qué firmas debe comprobar cada uno.

Se distinguen tres tipos de información: Información de seguridad vial, publicidad y conexión a internet. Tanto en las conexiones de seguridad vial como en las de publicidad, el vehículo que pertenece al grupo que recibe o produce la comunicación, se la envía al líder del grupo que la reenviará a su vez a todos los vehículos conectados del grupo y hacia las zonas donde el mensaje no haya sido difundido. Esta difusión la realizará utilizando los vehículos del grupo que estén mejor situados para la retransmisión o otros vehículos no pertenecientes al grupo que estén en el alcance de la conexión. En la figura 7 podemos ver una imagen que muestra las fases de conexión por las que pasa una conexión a internet mediante el reenvío de información. En este ejemplo en concreto el camión quiere realizar una conexión a internet, y para ello realiza una petición hacia un vehículo fuera de su grupo, el cual reenviará la petición hacia su líder. El líder enviará la petición hacia la RSU la cual le contestará dándole detalles

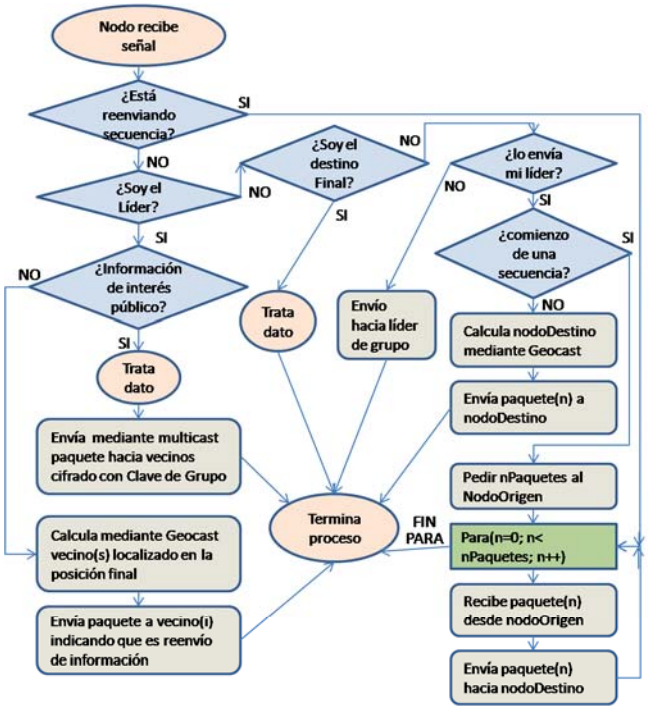


Fig. 6. Tratamiento de Mensajes en el Grupo

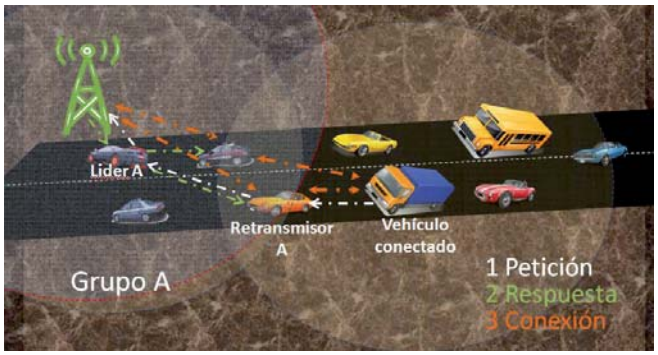


Fig. 7. Conexión a Internet

de la transmisión como número de paquetes para la conexión. Con esta información y sabiendo la posición y velocidad de los vehículos del grupo y del vehículo que se quiere conectar, el líder realiza los cálculos que le indicarán cuánto tiempo de conexión existe entre la RSU, los vehículos intermedios y el vehículo final. Después de esto el líder balanceará la carga de la conexión de forma que los paquetes lleguen al destino que los solicitó lo mejor y más rápidamente posible. Una vez que el líder haya avisado a los vehículos intermedios, estos vehículos se conectarán con la RSU y con el vehículo conectado y retransmitirán la conexión. Para que este esquema funcione se hace necesario el uso de mecanismos de cooperación como los expuestos en [6], debido a que sin este tipo de mecanismos los vehículos intermedios no tendrán los incentivos suficientes para retransmitir conexiones que no son suyas, imposibilitando de esta manera cualquier tipo de servicio que incorpore la conexión con las RSU.

VI. Conclusión

En este artículo se ha propuesto la utilización de grupos como solución para disminuir el número de comunicaciones que se producen en caso de tráfico denso y que provocan una disminución considerable en la calidad de las comunicaciones. Para ello se han descrito las diferentes fases del esquema completo para la incorporación de grupos en las VANETs. Se han diferenciado los estados en los que se puede centrar cada vehículo, desde el estado inicial en el que no pertenece a ningún grupo y procede a la detección de grupo, pasando por la elección de algún grupo ya creado, o la creación de un nuevo grupo, siendo en este caso el vehículo el líder de este nuevo grupo. También se contempla en el trabajo cómo actuar frente a las conexiones entrantes, debido a que la existencia de grupos afectan directamente a las comunicaciones que se producen en este tipo de redes.

Este trabajo está en desarrollo y en versiones futuras se incluirán resultados de simulación utilizando el simulador de tráfico SUMO y el de redes NS-2, analizando con detalle el funcionamiento de cada una de las fases y la reducción de conexiones con respecto a un mecanismo que no utilice grupos. Se estimarán los valores óptimos para los umbrales que definen cuándo merece la pena formar grupos y se analizarán los problemas relativos a la formación de grupos que se escapan utilizando modelos teóricos.

Agradecimientos

Investigación financiada por el Ministerio de Educación y Ciencia y la fundación Europea FEDER mediante el proyecto TIN2008-02236/TSI, y por la Agencia Canaria de Investigación, Innovación y Sociedad de la Información mediante el proyecto PI2007/005.

References

[1] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil and C. Hernández-Goya, Flexible Authentication in Vehicular Ad hoc Networks, Proceedings of APCC IEEE Asia Pacific Conference on Communications. 2009.

[2] P. Fan, P. Sistla, P. C. Nelson, Theoretical analysis of a directional stability-based clustering algorithm for vanets. Vehicular Ad Hoc Networks 2008:80-81

[3] Y. Gunter, B. Wiegel, H. P. Gromann, Cluster-based Medium Access Scheme for VANETs. Proceedings of the 2007 IEEE.

[4] Y. Gunter, B. Wiegel, H. P. Gromann, Medium Access Concept for VANETs Based on Clustering. VTC Fall 2007:2189-2193

[5] Y. Hao, Y. Cheng, K. Ren, Distributed key management with protection against RSU compromise in group signature based VANETs, IEEE GLOBECOM 2008

[6] J. Molina-Gil , P. Caballero-Gil and C. Caballero-Gil, A Vision of Cooperation Tools for VANETs, Proceedings of the International Workshop on Data Security and PrivAcy in wireless Networks, WoWMoM, 2010.

[7] P. Papadimitratos, L. Buttyán, J.P. Hubaux, F. Kargl, A. Kung, and M. Raya, Architecture for Secure and Private Vehicular Communications, 7th International Conference on ITS Telecommunications, 2007.

[8] M. Raya, P. Papadimitratos, and J.P. Hubaux, Securing Vehicular Communications - Assumptions, Requirements and Principles, Proceedings of Fourth Workshop on Embedded Security in Cars (ESCAR), 2006.

[9] S. Sivagurunathan, P. Subathra, V. Mohan and N. Ramaraj, Authentic Vehicular Environment Using a Cluster Based Key Management. EJSR, 2009:299-307

[10] Z. Y. Rawashdeh, S. Masud Mahmud, Media Access Technique for Cluster-Based Vehicular Ad Hoc Networks. VTC Fall 2008

Adaptación de una prueba de mezcla de votos para su uso con la cifra ElGamal

Víctor Mateu, Josep M. Miret, Francesc Sebé
Departamento de Matemática
Universitat de Lleida
C/ Jaume II, 69. E-25001 Lleida
Email: {vmateu,miret,fsebe}@matematica.udl.cat

***Resumen*—La parte más compleja de un sistema de votación electrónica basado en el paradigma de la mezcla de votos corresponde a la prueba de conocimiento nulo necesaria para demostrar que el proceso de mezcla se ha realizado correctamente. En este trabajo estudiamos la prueba propuesta por Peng, Boyd y Dawson en 2005 diseñada para sistemas donde los votos se cifran utilizando un criptosistema de clave pública homomórfico aditivo. De nuestro estudio se deduce que dicha prueba también puede ser utilizada cuando los votos se cifran utilizando la cifra ElGamal en su modo habitual (homomórfico multiplicativo) reduciéndose significativamente el tiempo necesario para su ejecución. Las mismas conclusiones se pueden aplicar a la propuesta mejorada y corregida del sistema publicada por Peng y Bao en 2009.**

I. Introducción

Un sistema de *votación electrónica remota* permite a sus participantes ejercer el derecho a voto a través de una red de comunicaciones, con lo cual se elimina la necesidad de desplazarse físicamente al colegio electoral. Un sistema de este tipo debe ofrecer suficientes propiedades de seguridad para garantizar que la votación no pueda ser manipulada de ningún modo. Mediante estas propiedades se pretende que solamente puedan votar (una única vez) los participantes que aparecen en el censo, de forma secreta y con garantías de que el resultado del recuento sea correcto. Además, el contenido de los votos ha de mantenerse en secreto hasta que la votación concluya.

Los sistemas de votación electrónica remota pueden clasificarse en tres paradigmas:

- Sistemas basados en firma ciega
- Sistemas con recuento homomórfico
- Sistemas con mezcla de votos

En el paradigma basado en firma ciega [8], los participantes, en una primera fase, interactúan con una entidad de confianza que firma su voto de forma ciega. Posteriormente, este voto que ha sido firmado ciegamente es enviado al colegio electoral virtual a través de un canal anónimo. Al finalizar la votación, todos los votos serán descifrados antes de proceder a su recuento. Este paradigma es el más simple y eficiente desde el punto de vista computacional. Desafortunadamente, una entidad de confianza deshonesta sería capaz de enviar votos falsos que resultarían indistinguibles de los legítimos. Por este motivo, es un paradigma que resulta poco aconsejable a pesar de haber sido implementado en diversas plataformas [5], [9].

En los sistemas con recuento homomórfico [4] los participantes envían su voto cifrado mediante un criptosistema homomórfico aditivo. El colegio, una vez ha recibido todos los votos, los suma homomórficamente obteniendo así un criptograma que al descifrarse da como resultado la suma de todos los votos individuales. Para que un sistema de este tipo sea seguro es necesario que los participantes, cuando envían su voto cifrado, demuestren que el criptograma enviado contiene un voto válido. Las pruebas de conocimiento nulo que permiten demostrar la validez del voto cifrado [15] tienen un coste que aumenta mucho cuando el número de opciones o candidatos es grande. Por tanto, este paradigma resulta apropiado únicamente para votaciones simples (por ejemplo, votaciones de tipo "Sí"/"No"). La plataforma [1] utiliza este paradigma.

El paradigma basado en la mezcla de votos [2] es el que más se asemeja a una votación tradicional con urna. En este caso, los participantes envían sus votos cifrados al colegio electoral virtual donde serán almacenados. Una vez se han recibido todos, los votos son mezclados y *recifrados* con el objetivo de que se pierda el vínculo entre voto y votante (privacidad). Finalmente, los criptogramas resultantes del proceso de mezcla son descifrados. La mezcla debe ser verificable en el sentido de que tiene que ser posible demostrar que durante el proceso de permutación y recifrado no se ha eliminado, añadido ni modificado ningún voto. En la literatura existen multitud de propuestas de sistemas que permiten verificar la corrección de un proceso de mezcla de votos [3], [11], [12], [16], [17], [18].

La gran ventaja de los sistemas basados en mezcla de votos es que con ellos se puede llevar a cabo cualquier tipo de votación siempre que un voto sea codificable mediante un mensaje que permita ser cifrado mediante un criptosistema de clave pública. Esto se debe a que después del proceso de mezcla todos los votos son descifrados y, en caso de hallarse uno con formato incorrecto (voto nulo), éste podrá ser descartado sin perjudicar el correcto desarrollo del proceso. De este modo, no es necesario que cada votante demuestre que el voto que ha enviado es correcto, evitando así la limitación de los sistemas homomórficos.

Existen propuestas híbridas, por ejemplo [14], que consiguen reducir el coste de verificar un proceso de mezcla aprovechando las propiedades homomórficas de un criptosistema.

I-A. Contribución y estructura del artículo

En [17] se presenta un sistema de votación electrónica basado en el paradigma de la mezcla de votos. La prueba allí presentada se halla descrita para votos cifrados utilizando el criptosistema (homomórfico aditivo) de Paillier. Los autores afirman que el sistema también puede usarse utilizando la cifra ElGamal en modo aditivo remarcando que la operación de descifrado requiere el cálculo de un logaritmo discreto (solamente aceptable cuando el espacio de textos en claro es reducido). En este artículo demostramos que la prueba [17] también puede usarse en el modo habitual de ElGamal (homomórfico multiplicativo). Además, se demuestra experimentalmente que el coste computacional de la prueba implementada sobre la cifra ElGamal es menor que sobre la cifra de Paillier. Los resultados de este trabajo son igualmente aplicables a la versión mejorada y corregida de dicho sistema presentada en [16].

El artículo se estructura de la siguiente forma. La primera sección es una introducción a la votación electrónica. En la sección II se describen los fundamentos criptográficos y en la sección III se presenta la prueba de Peng, Boyd y Dawson tal como se halla descrita en [17]. En la sección IV se detalla la prueba anterior una vez ha sido adaptada para su uso con la cifra ElGamal. Finalmente, la sección V está dedicada a los resultados experimentales.

II. Preliminares

II-A. La cifra Paillier

El criptosistema de Paillier [13] basa su seguridad en la intratabilidad del problema computacional de encontrar un enésimo residuo en $\mathbb{Z}_N$ con $N = p \cdot q$ cuando p y q son números primos no conocidos. La clave privada se genera escogiendo dos primos p y q. La clave pública está formada por $N = p \cdot q$ y un elemento $g \in \mathbb{Z}^*_{N^2}$ cuyo orden es múltiplo de N.

Un mensaje $m \in \mathbb{Z}^*_N$ se cifra calculando

$$c = g^m \cdot r^N \pmod{N^2}$$

donde $r \in \mathbb{Z}^*_N$ es un valor aleatorio.

Un criptograma c se descifra mediante la operación

$$m = L(c^\lambda \pmod{N^2}) \cdot \mu \pmod{N}$$

donde $L(u) = \frac{u-1}{N}$, $\lambda = \text{mcm}(p-1, q-1)$ y

$$\mu = (L(g^\lambda \pmod{N^2}))^{-1} \pmod{N}.$$

La cifra de Paillier es un homomorfismo aditivo. Dados dos criptogramas $c_1 = g^{m_1} \cdot r_1^N$ y $c_2 = g^{m_2} \cdot r_2^N$, vemos que $c = c_1 c_2 \pmod{N^2} = g^{m_1+m_2} \cdot (r_1 \cdot r_2)^N \pmod{N^2}$ de tal modo que al descifrar c obtendremos $D(c) = m_1 + m_2 \pmod{N}$.

La propiedad anterior permite que, dado un criptograma c, tomando r al azar, obtengamos un criptograma distinto $c' = c \cdot r^N \pmod{N^2}$ tal que $D(c) = D(c')$.

Tambien es fácil ver que dado un criptograma $c = g^m \cdot r^N \pmod{N^2}$ y un entero k, calculando $c' = c^k \pmod{N^2}$, se obtiene que $D(c') = k \cdot D(c) \pmod{N}$.

II-B. La cifra ElGamal

La cifra ElGamal [6] se basa en la dificultad de resolver el problema del logaritmo discreto planteado sobre un subgrupo G de orden primo q del grupo multiplicativo $\mathbb{Z}^*_p$. Dado un generador g de G, una clave privada se genera escogiendo un entero aleatorio x del intervalo $[1, q-1]$. La clave pública correspondiente a x es $y = g^x \pmod{p}$.

Dado un mensaje codificado como un elemento $m \in G$, éste se cifra calculando la tupla

$$(c, d) = (g^r, m \cdot y^r)$$

donde $r \in [1, q-1]$ es un valor aleatorio. A partir del conocimiento de la clave privada x, el texto en claro de un criptograma se recupera mediante la operación $m = \frac{d}{c^x}$.

La cifra ElGamal es un homomorfismo multiplicativo. Es fácil comprobar que dados dos criptogramas $(c_1, d_1) = (g^{r_1}, m_1 \cdot y^{r_1})$ y $(c_2, d_2) = (g^{r_2}, m_2 \cdot y^{r_2})$ que cifran los mensajes en claro m_1 y m_2 respectivamente, el criptograma $(c, d) = (c_1 \cdot c_2, d_1 \cdot d_2) = (g^{r_1+r_2}, m_1 \cdot m_2 \cdot y^{r_1+r_2})$ cifra el mensaje $m_1 \cdot m_2$.

Esta propiedad permite transformar un texto cifrado $(c, d) = (g^r, m \cdot y^r)$ en un criptograma distinto $(c', d') = (c \cdot g^{r'}, d \cdot y^{r'})$, con r' escogido al azar, cuyo texto en claro es el mismo que el de (c, d).

Finalmente, dado un criptograma (c, d) que cifra un texto en claro m, el criptograma $(c, k \cdot d)$ cifra el texto en claro $k \cdot m$.

II-C. La cifra ElGamal en modo homomórfico aditivo

La cifra ElGamal puede modificarse para que tenga la propiedad de ser homomórfica aditiva. Para ello, en lugar de cifrar un mensaje m directamente, se debe cifrar el valor $g^m \pmod{p}$.

Así, dados dos criptogramas $(c_1, d_1) = (g^{r_1}, g^{m_1} \cdot y^{r_1})$ y $(c_2, d_2) = (g^{r_2}, g^{m_2} \cdot y^{r_2})$, podemos operarlos y obtener $(c, d) = (c_1 \cdot c_2, d_1 \cdot d_2) = (g^{r_1+r_2}, g^{m_1 \cdot m_2} \cdot y^{r_1+r_2})$. Al descifrar el criptograma (c, d) se obtiene $h = g^{m_1+m_2}$ de donde se puede recuperar $m_1 + m_2$ resolviendo el logaritmo discreto $m_1 + m_2 = \log_g h$.

Debido a que para descifrar un mensaje es necesario calcular un logaritmo discreto, esta variante solamente es utilizable cuando el espacio de textos en claro posibles es reducido.

II-D. Mezcla verificable de votos

En un sistema de votación electrónica basado en el paradigma de la mezcla de votos, el colegio electoral virtual recibe los votos cifrados de todos los participantes y genera el conjunto $C = \{c_1, c_2, ..., c_n\}$ donde c_i es el criptograma que cifra el voto enviado por el i-ésimo participante.

En el proceso de mezcla, el colegio, a partir de una permutación aleatoria π de n elementos, genera un nuevo conjunto $C' = \{c'_1, c'_2, ..., c'_n\}$ tal que $D(c'_i) = D(c_{\pi(i)})$. Para garantizar la privacidad es necesario que sin conocimiento de la clave privada no sea posible vincular a qué voto de C corresponde cada voto de C', es decir, no ha de ser posible estimar la permutación π. Una vez se ha mezclado, para

proceder al recuento, simplemente se deben descifrar todos los votos de C' sin riesgo de que sea posible vincularlos con su origen.

Durante este proceso, a parte de la privacidad, también es necesario que se garantice la integridad del proceso. El colegio, a parte de permutar y recifrar los votos, también tiene que demostrar que este proceso se ha realizado correctamente. La prueba de corrección debe garantizar que el conjunto C' se ha generado correctamente (no se ha producido la manipulación de ningún voto).

III. El sistema de Peng, Boyd y Dawson

En [17] se presenta una prueba que permite demostrar que un proceso de mezcla de votos (cifrados mediante la cifra Paillier) se ha realizado correctamente. En esta prueba interactúan dos entidades. El probador (colegio electoral virtual) demuestra a un verificador (cualquier participante o supervisor de la elección) que el proceso de mezcla se ha realizado correctamente.

En esta prueba, el conjunto de entrada $C = \{c_1, \ldots, c_n\}$ está compuesto de criptogramas cifrados con la cifra de Paillier. Para generar el conjunto C' el probador procede como sigue:

1. Escoge aleatoriamente $r_i \in \mathbb{Z}_N^*$ para $i = 1, \ldots, n$.
2. Genera una permutación aleatoria π de n elementos.
3. Calcula $c'_i = c_{\pi(i)} \cdot r_i^N \pmod{N^2}$.
4. Publica el conjunto de votos mezclados y recifrados $C' = \{c'_1, \ldots, c'_n\}$.

A partir de este momento, el probador y el verificador ejecutan un protocolo a través del cual el probador demuestra al verificador que el contenido (texto en claro) de los textos cifrados del conjunto C' coincide con el contenido de los de C. La prueba se basa en que, a partir de ciertos valores s_i, s'_i escogidos por el verificador, el probador demuestre, en conocimiento nulo, que conoce unos valores secretos t_i, t'_i, para $i = 1, \ldots, n$, tales que,

$$\sum_{i=1}^{n} s_i D(c_i) = \sum_{i=1}^{n} t_i D(c'_i) \pmod{N},$$

$$\sum_{i=1}^{n} s'_i D(c_i) = \sum_{i=1}^{n} t'_i D(c'_i) \pmod{N},$$

$$\sum_{i=1}^{n} s_i s'_i D(c_i) = \sum_{i=1}^{n} t_i t'_i D(c'_i) \pmod{N},$$

donde $D(\cdot)$ denota la operación de descifrado.

A continuación se detalla el proceso para llevar a cabo la prueba en su forma no interactiva:

1. El verificador publica $s_i, s'_i \in [0, \ldots, 2^L - 1]$ para $i = 1, \ldots, n$ (L es un parámetro de seguridad).
2. El probador escoge aleatoriamente $r'_i \in \mathbb{Z}_N^*$ para $i = 1, \ldots, n$ y publica
$$c''_i = c_i'^{t_i} \cdot r_i'^N \pmod{N^2}$$
donde $t_i = s_{\pi(i)}$.
3. El probador calcula los siguientes valores:
$$R_1 = \prod_{i=1}^{n} r_i^{t_i} \cdot r'_i \pmod{N^2},$$
$$R_2 = \prod_{i=1}^{n} r_i^{t'_i} \pmod{N^2},$$
$$R_3 = \prod_{i=1}^{n} r_i^{t_i \cdot t'_i \pmod{\phi(N)}} \cdot r_i''^{t'_i} \pmod{N^2},$$
y publica:
$$C_1 = \frac{\prod_{i=1}^{n} c''_i}{\prod_{i=1}^{n} c_i'^{s_i}} \pmod{N^2},$$
$$C_2 = \prod_{i=1}^{n} c_i^{s'_i} \pmod{N^2},$$
$$C_3 = \prod_{i=1}^{n} c_i^{s_i \cdot s'_i \pmod{\phi(N)}} \pmod{N^2}.$$
4. El probador genera aleatoriamente W_1, W_2, W_3 y $v_i, v'_i, x_i \in \mathbb{Z}_N^*$, para $i = 1, 2, ..., n$.
5. El probador calcula
$$a_i = c_i'^{v_i} \cdot x_i^N \pmod{N^2}, \text{ para } i = 1, \ldots, n,$$
$$F = W_1^N \pmod{N^2},$$
$$A = \frac{\prod_{i=1}^{n} c_i'^{v'_i}}{W_2^N} \pmod{N^2},$$
$$B = \frac{\prod_{i=1}^{n} c_i''^{v'_i}}{W_3^N} \pmod{N^2}.$$
6. El probador calcula
$$c = \mathcal{H}(F, A, B, a_1, a_2, ..., a_n),$$
donde $\mathcal{H}$ es una función hash con una salida de 128 bits.
7. El probador calcula
$$z_1 = W_1 \cdot R_1^c \pmod{N^2},$$
$$z_2 = \frac{W_2}{R_2^c} \pmod{N^2},$$
$$z_3 = \frac{W_3}{R_3^c} \pmod{N^2},$$
$$\alpha_i = x_i \cdot r_i^c \pmod{N^2},\ i = 1, \ldots, n,$$
$$\gamma_i = v_i + c \cdot t_i \pmod{N},\ i = 1, \ldots, n,$$
$$\gamma'_i = c \cdot t'_i - v'_i \pmod{N},\ i = 1, \ldots, n.$$
8. El probador publica z_1, z_2, z_3, $\alpha_1, \cdots, \alpha_n$, $\gamma_1, \cdots, \gamma_n$, $\gamma'_1, \cdots, \gamma'_n$.
9. El verificador comprueba que
$$c = \mathcal{H}\left(\frac{z_1^N}{C_1^c}, \frac{C_2^c}{z_2^N \cdot \prod_{i=1}^{n} c_i'^{\gamma'_i}}, \frac{C_3^c}{z_3^N \cdot \prod_{i=1}^{n} c_i''^{\gamma'_i}}, \frac{c_1'^{\gamma_1} \cdot \alpha_1^N}{c_1''^c}, ..., \frac{c_n'^{\gamma_n} \cdot \alpha_n^N}{c_n''^c}\right).$$

La prueba presentada permite verificar que una mezcla de votos cifrados con la cifra de Paillier se ha realizado correctamente (véase [17]). Se puede comprobar que el coste computacional de la prueba es lineal respecto al número de votos n.

IV. Prueba adaptada para su uso con ElGamal

En la sección anterior se ha descrito la prueba tal como se detalla en [17]. Esa prueba requiere que los votos se hallen cifrados utilizando un criptosistema homomórfico aditivo. Puede parecer (y así se afirma en [17]), que si deseamos usarla con la cifra ElGamal, deberemos utilizar esta cifra en su modo homomórfico aditivo con lo cual, a la hora de descifrar cada voto será necesario resolver un

problema del logaritmo discreto. Para que este problema sea tratable se debe cumplir que el espacio de textos en claro sea pequeño. El problema es que con esta limitación se pierde la principal ventaja que ofrecen los sistemas de votación electrónica basados en la mezcla de votos frente a los sistemas con recuento homomórfico.

Un análisis más detallado permite llegar a la conclusión de que no es así. En ElGamal aditivo, para cifrar un mensaje m_1, primero calculamos $g_1 = g^{m_1}$ y después, este valor g_1 se cifra utilizando la cifra ElGamal en su modo habitual. Al descifrar, primero se obtiene g_1 y después se calcula $m_1 = \log_g g_1$.

Con ElGamal en modo aditivo, una prueba que demuestre que el conjunto de mensajes obtenido al descifrar es una permutación de los textos en claro $M = \{m_1, \ldots, m_n\}$ que se han recibido cifrados también demuestra que los valores obtenidos antes de resolver el logaritmo discreto son una permutación de $\{g^{m_1}, \ldots, g^{m_n}\}$.

Así pues, para utilizar la prueba [17] en un criptosistema homomórfico multiplicativo, lo único que se debe hacer es cifrar los votos utilizando el modo multiplicativo normal y después implementar la prueba como si la cifra se estuviese utilizando en modo aditivo.

A continuación se detalla como utilizar la prueba [17] teniendo en cuenta el razonamiento anterior cuando los votos se hallan cifrados con ElGamal en modo multiplicativo.

Ahora, el conjunto de votos recibidos está formado por criptogramas de la cifra ElGamal $C = \{(c_1, d_1), \ldots, (c_n, d_n)\}$.

La mezcla y recifrado de votos se realiza de la siguiente forma:

1. El probador escoge aleatoriamente $r_i \in \mathbb{Z}_q$ para $i = 1, \ldots, n$.
2. Genera una permutación π de n elementos.
3. Calcula $(c'_i, d'_i) = (c_{\pi(i)} \cdot g^{r_i}, d_{\pi(i)} \cdot y^{r_i})$ para $i = 1, \ldots, n$.
4. Publica el conjunto de votos mezclados y recifrados $C' = \{(c'_1, d'_1), \ldots, (c'_n, d'_n)\}$.

A continuación detallamos como realizar la prueba de [17] cuando los votos están cifrados utilizando ElGamal:

1. El verificador publica $s_i, s'_i \in [0, \ldots, 2^L - 1]$, para $i = 1, 2, \ldots, n$ (L es un parámetro de seguridad).
2. El probador escoge aleatoriamente $r'_i \in \mathbb{Z}_q$, para $i = 1, \ldots, n$, y publica $(c''_i, d''_i) = (c'^{t_i}_i \cdot g^{r'_i}, d'^{t_i}_i \cdot y^{r'_i})$ donde $t_i = s_{\pi(i)}$.
3. El probador calcula:

$$
\begin{aligned}
R_1 &= \textstyle\sum_{i=1}^{n}(r_i \cdot t_i + r'_i) \pmod q,\\
R_2 &= \textstyle\sum_{i=1}^{n} r_i \cdot t'_i \pmod q,\\
R_3 &= \textstyle\sum_{i=1}^{n} r_i \cdot t_i \cdot t'_i + r'_i \cdot t'_i \pmod q,
\end{aligned}
$$

y publica:

$$
\begin{aligned}
C_1 &= \frac{\prod_{i=1}^{n} c''_i}{\prod_{i=1}^{n} c'^{s_i}_i} \pmod p,\\
C_2 &= \prod_{i=1}^{n} c_i^{s'_i} \pmod p,\\
C_3 &= \prod_{i=1}^{n} c_i^{s_i \cdot s'_i \pmod q} \pmod p,\\
D_1 &= \frac{\prod_{i=1}^{n} d''_i}{\prod_{i=1}^{n} d'^{s_i}_i} \pmod p,\\
D_2 &= \prod_{i=1}^{n} d_i^{s'_i} \pmod p,\\
D_3 &= \prod_{i=1}^{n} d_i^{s_i \cdot s'_i \pmod q} \pmod p.
\end{aligned}
$$

4. El probador genera aleatoriamente W_1, W_2, W_3 y $v_i, v'_i, x_i \in \mathbb{Z}_q$, para $i = 1, \ldots, n$.
5. El probador calcula

$$
\begin{aligned}
a_i &= c'^{v_i}_i \cdot g^{x_i} \pmod p \text{ para } i = 1, \ldots, n,\\
b_i &= d'^{v_i}_i \cdot y^{x_i} \pmod p \text{ para } i = 1, \ldots, n,\\
F_c &= g^{W_1} \pmod p,\\
F_d &= y^{W_1} \pmod p,\\
A_c &= \frac{\prod_{i=1}^{n} c'^{v'_i}_i}{g^{W_2}} \pmod p,\\
A_d &= \frac{\prod_{i=1}^{n} d'^{v'_i}_i}{y^{W_2}} \pmod p,\\
B_c &= \frac{\prod_{i=1}^{n} c''^{v'_i}_i}{g^{W_3}} \pmod p,\\
B_d &= \frac{\prod_{i=1}^{n} d''^{v'_i}_i}{y^{W_3}} \pmod p.
\end{aligned}
$$

6. El probador calcula

$$c = \mathcal{H}(F_c, F_d, A_c, A_d, B_c, B_d, a_1, b_1, ..., a_n, b_n),$$

donde $\mathcal{H}$ es una función hash con salida de 128 bits.
7. El probador calcula

$$
\begin{aligned}
z_1 &= W_1 + R_1 \cdot c \pmod q,\\
z_2 &= W_2 - R_2 \cdot c \pmod q,\\
z_3 &= W_3 - R_3 \cdot c \pmod q,\\
\alpha_i &= x_i + r_i \cdot c \pmod q,\ i = 1, \ldots, n,\\
\gamma_i &= v_i + c \cdot t_i \pmod q,\ i = 1, \ldots, n,\\
\gamma'_i &= c \cdot t'_i - v'_i \pmod q,\ i = 1, \ldots, n.
\end{aligned}
$$

8. El probador publica z_1, z_2, z_3, $\alpha_1, \ldots, \alpha_n$, $\gamma_1, \ldots, \gamma_n$, $\gamma'_1, \ldots, \gamma'_n$.
9. El verificador comprueba:

$$
\begin{aligned}
c = \mathcal{H}\Bigg(& \frac{g^{z_1}}{C_1^c}, \frac{y^{z_1}}{D_1^c}, \frac{C_2^c}{g^{z_2} \cdot \prod_{i=1}^{n} c_i^{\gamma'_i}}, \frac{D_2^c}{y^{z_2} \cdot \prod_{i=1}^{n} d_i^{\gamma'_i}},\\
& \frac{C_3^c}{g^{z_3} \cdot \prod_{i=1}^{n} c_i^{\gamma'_i}}, \frac{D_3^c}{y^{z_3} \cdot \prod_{i=1}^{n} d_i^{\gamma'_i}},\\
& \frac{c'^{\gamma_1}_1 \cdot g^{\alpha_1}}{c''^{c}_1}, \frac{d'^{\gamma_1}_1 \cdot y^{\alpha_1}}{d''^{c}_1}, ..., \frac{c'^{\gamma_n}_n \cdot g^{\alpha_n}}{c''^{c}_n}, \frac{d'^{\gamma_n}_n \cdot y^{\alpha_n}}{d''^{c}_n} \Bigg).
\end{aligned}
$$

V. Resultados experimentales

La prueba [17] ha sido implementada en Java [7] para votos cifrados con la cifra de Paillier (descrita en la sección III) y para votos cifrados con ElGamal multiplicativo (descrita en la sección IV). Se han realizado diferentes experimentos variando el nivel de seguridad y el número de votos. Para tener un mismo nivel de seguridad en las dos cifras se debe escoger el

número $N = p \cdot q$ de Paillier y el módulo p de ElGamal de igual longitud.

	Tiempo (s)	
Votos	Paillier	ElGamal
600	16033	6017
1000	30915	11817
1400	96509	36636
1800	136378	51070

Cuadro I
TIEMPO EN SEGUNDOS NECESARIO PARA PROBAR LA CORRECCION DE UNA MEZCLA DE VOTOS USANDO CLAVE DE 256 BITS.

A partir del cuadro I (donde se han usado claves de 256 bits), se puede comprobar que el tiempo de ejecución aumenta linealmente con el número de votos tanto si se cifra con Paillier como ElGamal. La relación de tiempos entre las dos cifras muestra que la prueba, cuando se trabaja con ElGamal, es aproximadamente tres veces más rápida que con Paillier.

	Tiempo (s)	
Votos	Paillier	ElGamal
600	79625	17705
1000	145136	31862
1400	447886	95779
1800	641608	131665

Cuadro II
TIEMPO EN SEGUNDOS NECESARIO PARA PROBAR LA CORRECCION DE UNA MEZCLA DE VOTOS USANDO CLAVE DE 512 BITS.

El cuadro II muestra el mismo experimento utilizando claves de 512 bits. Con esta longitud de clave, la prueba implementada sobre la cifra ElGamal resulta ser cinco veces más rápida. Finalmente, el cuadro III resume los resultados con claves de 1024 bits. En este caso la prueba con ElGamal resulta ser unas nueve veces más rápida.

	Tiempo (s)	
Votos	Paillier	ElGamal
600	463904	58090
1000	820066	99728
1400	1211291	143156
1800	1639864	186634

Cuadro III
TIEMPO EN SEGUNDOS NECESARIO PARA PROBAR LA CORRECCION DE UNA MEZCLA DE VOTOS USANDO CLAVE DE 1024 BITS.

A partir de los resultados experimentales se observa que la prueba implementada sobre la cifra ElGamal es considerablemente más rápida que cuando se implementa sobre Paillier. La diferencia en el tiempo de ejecución aumenta a medida que elevamos el nivel de seguridad. El motivo es que aunque los criptogramas de la cifra ElGamal están formados por dos componentes, en el momento de operar sobre ellos se realizan operaciones sobre $\mathbb{Z}_p$ mientras que con Paillier se trabaja en $\mathbb{Z}_{N^2}$, siendo p y N números de la misma longitud (con lo cual la longitud de N^2 dobla la de p).

El proceso de adaptación que se ha presentado en este trabajo puede aplicarse de forma análoga a la versión corregida y mejorada del sistema presentada en [16].

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia mediante los proyectos MTM2007-66842-C02-02 y "ARES" CSD2007-0004 y por la Generalitat de Catalunya mediante el proyecto 2009SGR-442.

REFERENCIAS

[1] P. Akritidis, Y. Chatzikian, M. Dramitinos, E. Michalopoulos, D. Tsigos, N. Ventouras, "The VoteSecure secure Internet voting system", *Lecture Notes in Computer Science*, vol. 3477, pp. 420–423, 2005.
[2] D. Chaum. "Untraceable electronic mail, return addresses and digital pseudonyms", *Communications of the ACM*, 24(2) pp. 84–88, 1981.
[3] J. Cichoń, M. Klonowski, M. Kutyłowski, "Distributed verification of mixing - local forking proofs model", *Lecture Notes in Computer Science*, vol. 5107, pp. 128–140, 2008.
[4] J.D. Cohen(Benaloh), M.J. Fischer, "A robust and verifiable cryptographically secure election scheme" *FOCS'85* pp. 372–382, 1985.
[5] L.F. Cranor, R.K. Cytron, "Design and implementation of a security-conscious electronic polling system", Washington University Computer Science Technical Report WUCS-96-02, 1996.
[6] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithm", *IEEE Trans. on Information Theory*. vol. 31, pp. 469–472, 1985.
[7] D. Flanagan, *Java in a nutshull (5th ed.).* O'Reilly & Associates, 2005.
[8] A. Fujioka, T. Okamoto, K. Ohta, "A practical secret voting scheme for large scale elections", *Lecture Notes in Computer Science*, vol. 718, pp. 244–251, 1993.
[9] S. Ibrahim, M. Kamat, M. Salleh, S.R.A. Aziz, "Secure E-voting with blind signature", *NCTT'03*, pp. 193-197, 2003.
[10] B. Lee, K. Kim, "Receipt-free electronic voting scheme with a tamper-resistant randomizer", *Lecture Notes in Computer Science*, vol. 2587, pp. 389 - 406, 2003.
[11] A. Neff, "Verifiable mixing (shuffling) of ElGamal pairs", `http://www.votehere.net/documentation/vhti`, 2004.
[12] L. Nguyen, R. Safavi-Naini, K. Kurosawa, "Verifiable shuffles: a formal model and a paillier-based efficient construction with provable security", *Lecture Notes in Computer Science*, vol. 3089, pp. 61–75, 2004.
[13] P. Paillier, "Public key cryptosystem based on composite degree residuosity classes", *Lecture Notes in Computer Science*, vol. 1592, pp. 223–238, 1999.
[14] K. Peng, "A hybrid e-voting scheme", *Lecture Notes in Computer Science*, vol. 5451, pp. 195 - 206, 2009.
[15] K. Peng, F. Bao, "Efficient vote validity check in homomorphic electronic voting", *Lecture Notes in Computer Science*, vol. 5461, pp. 202–217, 2009.
[16] K. Peng, F. Bao, "Correction, optimisation and secure and efficient application of PBD shuffling", *Lecture Notes in Computer Science*, vol.5487, pp. 425–437, 2009.
[17] K. Peng, C. Boyd, E. Dawson, "Simple and efficient shuffling with provable correctness and ZK privacy", *Lecture Notes in Computer Science*, vol.3621, pp. 188–204, 2005.
[18] F. Sebé, J. Miret, J. Pujolàs, J. Puiggalí, "Simple and efficient hash-based verifiable mixing for remote electronic voting", *Computer Communications*, vol. 33, pp. 667–675, 2010.

Estudio de los Sistemas de Verificación para Votaciones Electrónicas Presenciales

Roger Jardí Cedó, Jordi Pujol Ahulló y Jordi Castellà Roca
Departament d'Enginyeria Informàtica i Matemàtiques
UNESCO Chair in Data Privacy
Universitat Rovira i Virgili
Av. Països Catalans 26. E-43007 Tarragona, Spain
Email: {roger.jardi, jordi.pujol, jordi.castella}@urv.cat

***Resumen*—Las elecciones son una parte muy importante de una democracia. Mediante éstas, la sociedad puede elegir a sus representantes, tomar una decisión o expresar una opinión. Alguien podría estar interesado en manipular los resultados de dichas elecciones sin ser descubierto. Por este motivo es necesario un control del proceso de votación que detecte y evite cualquier tipo de alteración. En este trabajo se estudian los principales sistemas de verificación (académicos y comerciales), que realizan dicho control en los sistemas de votación electrónica presencial. Con este fin se ha diseñado un marco comparativo común con el que se analizan todos los sistemas de verificación de votación electrónica. Este marco comparativo agrupa diferentes propiedades, cubriendo importantes áreas tales como la interacción con el usuario o la seguridad.**

I. Introducción

Un proceso electoral consiste en elegir a una persona o partido, es decir, un representante de todos los miembros de una comunidad (p.e., una empresa, un estado o un país). Para un candidato esto supone una gran responsabilidad pero a la vez un gran poder (p.e., fondos, capacidad de cambiar leyes).

Por este motivo, existe la necesidad de comprobar la corrección del proceso de votación y de los resultados, asegurando así que los votos corresponden a las intenciones reales de los votantes. Puesto que debe mantenerse el secreto de voto y el anonimato del votante en todo momento, la verificación puede resultar difícil. A pesar de esto, cualquier sistema de votación debe proporcionar algún tipo de verificación.

Los sistemas de votación tradicionales usan papeletas electorales de papel como soporte para emitir el voto. Actualmente las nuevas tecnologías han propiciado la aparición de los sistemas de votación electrónica o "e-voting", que usan dispositivos electrónicos. Estos sistemas proporcionan ventajas respecto a los sistemas convencionales como una aceleración del proceso de escrutinio o un ahorro de papel con consecuencias económicas, logísticas y ecológicas. Las primeras iniciativas fueron en EE.UU., en 1964, donde se utilizaron tarjetas perforadas y máquinas de recuento, aunque más recientemente se han utilizado escáneres ópticos y técnicas criptográficas.

La aparición de estos nuevos sistemas ha hecho replantear un conjunto de problemáticas, ya conocidas en los sistemas antiguos, para solucionarlas y mejorarlas. Estas viejas problemáticas derivan de la seguridad, como es el caso del secreto de voto o el anonimato del votante. Éstas afectan directamente el proceso de verificación. Para ello se definen tres tipos de verificación: *individual*, *universal* y *extremo-a-extremo* (del inglés "end-to-end"). El primer tipo, definido desde el punto de vista del votante, permite que éste pueda comprobar que su voto ha sido emitido y contabilizado correctamente. La verificación universal permite comprobar que los votos de las urnas no han sido manipulados y que se han escrutado correctamente. Su objetivo es el de garantizar que el proceso que va desde la emisión del voto hasta el recuento se haya realizado correctamente. En los sistemas tradicionales (basados en papeletas en papel y procesos manuales), se realizan ambos tipos de verificación por medio de un conjunto de procesos manuales. En cambio, para realizar dichas verificaciones en sistemas de *e-voting* es necesaria una combinación de nuevas tecnologías y nuevos procesos. La verificación extremo-a-extremo es un tipo de verificación individual que permite comprobar que el voto ha sido emitido y contado correctamente en cualquier momento de la votación. Su objetivo es el de aumentar la confianza del votante en el sistema y, consecuentemente, en la veracidad e integridad de los resultados electorales. Los nuevos sistemas de votación, a diferencia de los tradicionales, facilitan este tipo de verificación.

Este estudio presenta un análisis de los sistemas de verificación de votaciones electrónicas (SVV de ahora en adelante) y muestra como éstos consiguen alguno de los tres tipos de verificación antes descritos. En particular, se centra el enfoque en los SVVs del tipo presencial, en inglés "poll site", ya que éstos son los más comunes actualmente. No obstante, en los siguientes capítulos, se verá la evolución que sufren los SVVs presenciales hacia el modelo remoto, adoptando modelos intermedios como el presencial-remoto. En este tipo de sistemas, el voto se emite de forma controlada, como en los presenciales, aunque el escrutinio se realiza remotamente, como en el tipo remoto (es decir, en un centro o sitio distinto al de la emisión). Los sistemas incluidos en este trabajo son los más relevantes del mundo comercial y académico de la última década. Por lo tanto, la contribución de este trabajo es doble: (i) la *definición de un marco de evaluación común* para una justa comparación de todos los sistemas y, (ii) el *estudio y comparación de los sistemas de votación electrónica* más importantes.

Estructura del documento. La siguiente sección introduce los conocimientos necesarios y las bases para entender el estudio. La Sección 3 presenta el marco de evaluación y la Sección 4 el análisis de todos los sistemas de verificación de votación (SVVs) incluidos. La Sección 5 analiza globalmente todos los sistemas, y por último, la Sección 6 presenta las conclusiones finales de este trabajo.

II. Conocimientos Previos

En este estudio se considera que el proceso de votación estándar está compuesto por tres fases: (i) el *registro del votante y su identificación*, (ii) la *emisión de voto* y (iii) el *escrutinio de votos*, donde todos los votos son contabilizados de manera segura y los resultados son imparciales y públicos. El proceso de votación incluye todos los procedimientos y tecnologías necesarios para confiar en la votación.

II-A. Tipos de Votación

A continuación se presenta la clasificación de los sistemas de votación según el lugar donde el votante emite el voto (Sección II-A1) y la clasificación que realiza la legislación de EE.UU. llamada "HAVA classification"(Sección II-A2) sobre los sistemas de verificación de votaciones (SVVs). Esta última se utilizará más adelante para organizar los sistemas de votación analizados.

II-A1. Clasificación según el entorno: Según el sitio de votación y su entorno inmediato, donde el votante puede ejercer su voto, los sistemas de votación se pueden clasificar en sistemas **presenciales** y **remotos**. Para emitir un voto en el *primer tipo*, los votantes tienen que ir a un sitio dedicado a este propósito (p.e., colegio electoral) y con algún tipo de control físico (entorno controlado). En los *sistemas remotos*, los votantes pueden votar desde un sitio lejano al sitio donde se efectúa el escrutinio. Los ejemplos más importantes son el **voto postal** y el **voto por Internet**.

Recientemente se ha propuesto un nuevo tipo de sistemas que pretende aprovechar las ventajas tanto de los sistemas presenciales como de los remotos; el **presencial remoto**. Este tipo de sistemas permiten emitir el voto en un entorno controlado, aunque su escrutinio se realiza en otro sitio destinado específicamente al recuento seguro y centralizado de todos los votos.

II-A2. Clasificación HAVA: Esta clasificación fue promovida por la *Election Assistance Commission* (EAC), una agencia independiente del Gobierno de EE.UU. creada por *Help America Vote Act* 2002 (HAVA). La agencia EAC definió las *Voluntary Voting System Guidelines* (VVSG) en 2005 [1], que consistían en unas directrices que clasifican los SVVs en cuatro tipos: (i) los **SVVs basados en separación de procesos** tienen una arquitectura modular que separa el sistema en dos procesos diferentes, la generación del voto, y su emisión; (ii) los **SVVs basados en evidencias** registran todas las acciones realizadas por los votantes durante la emisión del voto; (iii) los **SVVs directos o de verificación directa** generan un registro paralelo al del voto que permite la verificación directa del votante, es decir, que el votante no requiera de ninguna tecnología para interpretar el contenido del registro; finalmente, (iv) los **SVVs basados en criptografía extremo a extremo** usan métodos criptográficos para la construcción de recibos. Éstos permiten a los votantes verificar, sin comprometer su privacidad, que su voto no ha sido modificado.

III. Marco de Evaluación Común

Esta sección establece la clasificación que se utilizará para organizar los sistemas de verificación a estudiar y también define y clasifica las propiedades de éstos. Todo ello constituye el marco común que se usará más adelante para analizar y comparar los SVVs.

III-A. Clasificación de los SVVs

En esta sección se presentan los criterios la clasificación de los SVVs. El **año de** su primera **publicación** es la última propiedad usada para organizarlos.

1. Entre los sistemas **electrónicos** y los **basados en papel**, sólo se consideran los electrónicos. Éstos requieren que el voto esté en formato electrónico en vez de papel.
2. Se usa la *clasificación HAVA* antes mencionada para diferenciar los SVVs en sistemas basados en **separación de procesos**, en **evidencias**, en **criptografía extremo a extremo** (E2E) y **directos**.
3. Se distingue entre sistemas **integrales** e **independientes**. Los primeros son sistemas completos que realizan todo el proceso de votación, incluido el de verificación. En cambio, los independientes están diseñados únicamente para verificar la votación sin estar supeditados a un sistema de votación determinado.

III-B. Propiedades Evaluadas de los SVVs

A continuación se presentan las características consideradas importantes para evaluar los sistemas de verificación. Su clasificación obedece a los aspectos del proceso de votación al que conciernen: *interacción con el usuario*, *seguridad*, *integrabilidad* (con un sistema de votación existente) y *aspectos técnicos*.

Interacción con el usuario. La interacción con el votante determina en gran medida la impresión y la confianza que da el sistema:

1. **Accesibilidad.** Facilidad de uso del sistema (p.e., usuarios con algún tipo de discapacidad).
2. **Impacto de uso.** Complejidad del proceso de emisión de voto comparado con el proceso habitual.
3. **Confiabilidad.** Confianza en el proceso de votación desde el punto de vista de los votantes.

Seguridad. Las propiedades de seguridad están relacionadas con el *votante* y con el *proceso de votación*:

Relacionados con el votante:

4. **Secreto de voto.** El sistema impide que una tercera persona conozca el contenido del voto.
5. **Anonimato.** El sistema impide que se mantenga cualquier relación entre el voto y el votante.

6. **Resistencia a la coacción.** El sistema impide que el votante pueda probar a una tercera persona el contenido de su voto.
7. **Verificación individual.** El sistema permite que el votante pueda comprobar que su voto se ha contabilizado correctamente.

Relacionados con el proceso de votación:

- **Verificación universal.** Ésta se descompone en:
  8. **Integridad de la urna.** Únicamente los votos de votantes autorizados pueden estar en la urna de manera inalterada hasta el final del proceso de votación (antes del escrutinio).
  9. **Precisión del escrutinio.** El escrutinio debe contar todos los votos emitidos de manera correcta no antes del final de la votación.
10. **Auditabilidad.** El sistema de votación electrónica permite a una tercera persona, sin comprometer otras propiedades de seguridad, analizar lo sucedido en las elecciones.

Integración. El SVV, sea integral o independiente, tiene la posibilidad y la efectividad de adaptarse/integrarse con otros sistemas de votación, actuando el SVV de forma independiente (de la misma forma que se realizó en [2]). Se tiene en cuenta la sincronización de las operaciones, especialmente la emisión de votos, entre el sistema de votación y el SVV.

11. **Integración**. Facilidad de implementación/adaptación del sistema evaluado como sistema verificador independiente de otras infraestructuras de votación.
12. **Gestión de la información.** Evalúa cuando el subsistema de emisión de votos de los sistemas de votación y el sistema evaluado garantizan tanto atomicidad de operaciones, como resistencia a fallos (p.e., errores de usuarios, desconexiones de cables).

Aspectos técnicos. Se analiza el rendimiento de los SVVs desde un punto de vista técnico:

13. **Simplicidad.** Cuando el SVV es directo y sencillo.
14. **Disponibilidad.** El votante debe poder emitir su voto sólo cuando se le permita, durante el tiempo establecido para ello, y previniendo que pueda emitir más de un voto (si no está permitido).
15. **Escalabilidad.** El sistema verificador escala computacionalmente.
16. **Flexibilidad.** Esta propiedad evalúa el nivel de libertad que ofrece el sistema verificador (p.e., número de candidatos, modo escritura –"write-in").

Representación. Por simplicidad, a continuación se muestra la notación usada en cada una de las 16 propiedades evaluadas en los sistemas estudiados.

IV. Estudio y Comparación de los SVVs

En esta sección se presentan los SVVs (Sección IV-A), su análisis (Sección IV-B) y el estudio de las sinergias entre los sistemas de votación electrónica y las técnicas criptográficas (Sección IV-C).

Cuadro I
Lista de valores de las propiedades evaluadas.

Interacción con el Usuario	↑ / ↓ / ∼: Buena/Débil/Aceptable.
Seguridad	**S**/**N**/∼: **S**í/**N**o/Parcialmente.
Integración	**NT**: **N**o existen requisitos **T**écnicos adicionales (en terminales de votación, etc.). **T**: Existen requisitos **T**écnicos adicionales. **NSW**: **N**o existen requisitos **S**oft**W**are adicionales (en terminales de votación, etc.). **SW**: Existen requisitos **S**oft**W**are adicionales.
Gestión de la información	**NA**: **N**o existe **A**tomicidad en las operaciones. **A**: Existe **A**tomicidad en las operaciones. **PD**: Existe **P**érdida de **D**atos. **NPD**: **N**o existe **P**érdida de **D**atos.
Aspectos Técnicos	↑ / ↓: Propiedad Alta/Baja.
En cualquier caso	"N/A": La propiedad no fue considerada por el sistema.

IV-A. Presentación y Clasificación de los SVVs

Por razones de espacio, a continuación se listan brevemente los sistemas de mayor relevancia, y que han estado incluidos en el presente estudio. Para mayor detalle, se invita al lector a visitar las referencias externas indicadas.

Siguiendo la categorización HAVA, en el conjunto de *SVVs basados en separación de procesos* se incluye **Modular Voting Architecture ("Frog")** [3]. En la categoría de *SVVs basados en evidencias* se han estudiado **VVAATT** [4] y **VVVAT** [5]. Finalmente, en el grupo de *SVVs basados en criptografía extremo a extremo* se han analizado **VoteHere** [6], **VoteBox** [7], **Three-Ballot** [8] y **E-valg** [9].

IV-B. Análisis y Evaluación de los SVVs

Para analizar y valorar los SVVs se siguen las propiedades antes descritas en el marco común de evaluación. En la Cuadro III se puede ver el resumen de este análisis.

Interacción con el usuario. Dado que todos los SVVs usan DREs (del inglés "**d**irect **r**ecording **e**lectronic voting terminals") para emitir los votos, todos ellos tienen un cierto grado de **accesibilidad**. No obstante, algunos lo mejoran usando audioguías (VVAATT) o añadiendo otras tecnologías asistivas (como el ratón o el teclado) (VoteBox y E-valg). En el caso del E-valg, esto ya fue probado en los estudios [2], [10]. En cuanto al **impacto de uso**, sistemas como Frog, VoteBox y Three-Ballot presentan mayor complejidad y un proceso de votación ligeramente más largo. Por ejemplo, en el Frog existe una separación estricta entre los procesos de generación y emisión de voto (a pesar de que un votante pueda traer la papeleta llena de casa); VoteBox permite a los votantes realizar un "immediate ballot challenge" [11], que es una manera de alertar del mal funcionamiento del equipo de emisión e inutilizar el terminal de votación; o Three-Ballot que usa una multi-papeleta compuesta por tres partes. Además, todos ellos incorporan tres técnicas distintas con el fin de incrementar la **confianza** en el sistema: (i) *frogs* (Frog) y *recibos* (VoteHere, VoteBox, Three-Ballot y E-valg) son elementos tangibles para

el votante, (ii) *audio guías* (VVAATT), y *boletines* publicados en Internet (excepto VVAATT).

Seguridad. VVAATT/VVVAT no garantizan la confidencialidad de voto, dado que todas las grabaciones (audio o vídeo) muestran el orden *secuencial* de votación. Además, el equipo de grabación del VVAATT/VVVAT tiene una *protección* muy débil (ya que se accede a menudo) y unas técnicas de *extracción* de información de los registros de grabación poco fiables y lentas. A pesar que este sistema no garantiza el anonimato del votante y *no es fiable* en su conjunto, se ha incluido en el estudio por ser el referente de los sistemas basados en evidencias. A continuación se analizan el resto de sistemas.

Seguridad relacionada con el votante. A excepción de Frog, todos los sistemas usan una infraestructura de clave pública (PKI), la mayoría de ellos ElGamal, para garantizar el **secreto de voto**. Sin embargo, los SVVs usan técnicas muy diferentes para proteger el **anonimato del votante**. Mientras que Frog utiliza un simple algoritmo aleatorio, Three-Ballot separa cada una de las tres partes del voto y las almacena usando sus valores hash. También aparecen técnicas más complejas: mixing (VoteHere), criptografía homomórfica aditiva (VoteBox) o un esquema híbrido (E-valg), que combina criptografía homomórfica multiplicativa con mixing. El uso de esquemas umbral, extendido en gran parte de estos sistemas, previene ataques de seguridad procedentes de partes teóricamente confiables. Por otro lado, los sistemas que utilizan recibos pueden permitir la **coacción** y la **venta de votos**, ya que pueden reflejar la opción de voto en el resguardo. Para solucionar esta problemática se usan los códigos de retorno que impiden adivinar el voto; es el caso de VoteBox, Three-Ballot, E-valg y VoteHere. Sin embargo, este último puede tener un defecto, ya que el votante podría evidenciar su intención de voto dado que dispone del voto cifrado junto con los códigos de retorno [12]. Frog, a pesar de no utilizar recibos, tampoco cumple con dichas propiedades, ya que no asegura el secreto de voto. Además, los sistemas que usan recibos *mejoran* la **verificación individual** a *E2E*.

Seguridad relacionada con el sistema. A excepción de Frog, todos los sistemas garantizan la **integridad de la urna** mediante diferentes tecnologías (como pruebas de conocimiento nulo –*ZKPs*–, firmas digitales o esquemas umbral). Ver el Cuadro II para mayor detalle. VoteHere, VoteBox, Three-Ballot y E-valg avalan la **precisión del escrutinio**. Los sistemas homomórficos hacen un recuento más eficiente que las técnicas de mixing [13], [14]. En cuanto a la **auditabilidad**, Three-Ballot crea unos registros o pistas de auditoría para cualquier operación relacionada con los votantes, aunque ninguna sobre el proceso de recuento. Los sistemas evaluados que ofrecen más prestaciones en éste ámbito son VoteBox y E-valg, ya que usan trazas inmutables. VoteBox construye un sistema de auditoría totalmente distribuido, mientras que E-valg sólo audita los elementos críticos del sistema.

Integración. En general todos los SVVs evaluados tienen alguna dependencia, ya sea de software o hardware (ver Cuadro III para más detalles), para integrarlos en los DREs.

Cuadro II
TÉCNICAS DE SEGURIDAD USADAS POR LOS SVVS

		Técnicas de Seguridad			
		ZKPs	Firma Digital	Esquema Umbral	Sistema Auditoría
SVV	Frog	No	Sí	No	No
	VoteHere	Sí	Sí	Sí	No
	VoteBox	Sí	Sí	Sí	Sí
	Three-Ballot	No	Sí	No	No
	E-valg	Sí	Sí	Sí	Sí

Los únicos sistemas que implementan algún tipo de **gestión de datos** son VoteBox y E-valg [2], ya que garantizan la atomicidad del voto, son resistentes a pérdidas, y manipulación de datos.

Aspectos técnicos. VoteBox y Frog son más **complejos** que el resto de sistemas estudiados, puesto que tienen una estructura distribuida –el último debido a la separación de procesos. Sin embargo, VoteBox es el único que estructuralmente proporciona *replicación* de la información sensible, susceptible de ser objeto de algún tipo de ataque. Esta característica otorga al sistema un alto grado de **disponibilidad**. Otra buena propiedad de VoteBox es su **escalabilidad** (debida a su arquitectura), además de un escrutinio rápido de votos (como consecuencia del uso de criptografía homomórfica). VoteBox y E-valg comparten esta propiedad. A pesar de ello, ambos deben abordar el voto presencial remoto con cautela, garantizando la infraestructura necesaria a fin de no sobrecargar el sistema. Finalmente, sólo Frog, VoteHere y E-valg dan **flexibilidad** al tipo votación y al formato del voto. Por el contrario, VoteBox, que usa las propiedades homomórficas *aditivas* de ElGamal, no soporta tipos de elecciones muy complejas. Three-Ballot sólo es adecuado para papeletas formadas por tres partes, a pesar de que el contenido de la papeleta es flexible.

IV-C. Estudio de Tendencias en SVVs

Del análisis anterior se extraen *tres tendencias claras* relativas a los siguientes aspectos: (i) ámbito de votación, (ii) tecnología de la votación y (iii) grado de verificabilidad.

Estudio del ámbito de votación. En este estudio se ha analizado SVVs *presenciales*, y todos ellos usan DREs como terminales de votación. Claramente, los DREs resultan de gran utilidad, puesto que permiten la gestión electrónica de los votos. En este sentido cabe destacar la tendencia existente de migración de sistemas electorales *presenciales* a *presenciales remotos*. Dicha tendencia es consecuencia no sólo de la tecnología, sino también de la evolución natural de las reglas democráticas. A pesar de ello, mientras que VoteBox se *adaptó* para soportar esquemas de voto presenciales remotos, E-valg *fue estructuralmente diseñado* para tal propósito.

Estudio de la tecnología de votación. En este caso se considera la tecnología de votación usada desde la emisión de los votos hasta su escrutinio, por lo que los sistemas VVAATT/VVVAT no se incluyen en este estudio. El objetivo de estas tecnologías es el de proveer seguridad (tales como anonimato, integridad de la urna o precisión del escrutinio). La tendencia que presentan estas soluciones, de las más simples

a las más complejas y fiables, es tal como sigue. Mientras que Frog usa meros algoritmos *aleatorios* en el tratamiento de los votos para garantizar el *anonimato de los votantes*, VoteHere usa técnicas de *mixing* más fiables. VoteBox y Three-Ballot usan criptografía *homomórfica aditiva* (que es computacionalmente intensiva) para garantizar el *anonimato*, a la par que facilitan el *escrutinio*. La tecnología más compleja, pero más flexible y fiable, es la usada por E-valg: el *esquema híbrido*. Dicho esquema se compone de criptografía homomórfica multiplicativa (menos intensiva computacionalmente que la aditiva [13], [14]) y mecanismos de mixing. Claramente, estas tecnologías de votación llegan a un *compromiso* entre (i) garantizar que sean más seguras y fiables, y (ii) asegurar que sean rápidas y eficientes en el uso de recursos. Es claro que esta tendencia desde simples técnicas de aleatoriedad a esquemas híbrido es una consecuencia directa de la continua permeabilidad de los sistemas de votación con respecto a los últimos avances criptográficos.

Estudio de la verificabilidad. Los sistemas analizados se organizan como sigue. (i) Sistemas basados en VVAATT/VVVAT y Frog proveen una verificabilidad deficiente y básica respectivamente de los procesos de votación. Principalmente se centran en garantizar únicamente un cierto grado de verificación individual y desatendiendo la universal y la E2E. (ii) VoteHere y Three-Ballot ofrecen un grado aceptable de verificabilidad (individual, universal y E2E). Finalmente, (iii) E-valg y VoteBox aseguran un buen nivel de verificabilidad, a la vez que definen un sistema de auditoría robusto.

En conclusión, VoteBox y E-valg son las mejores alternativas estudiadas como sistemas de votación. No obstante, E-valg ofrece unas características que lo hacen atractivo. La razón de ello es porque E-valg provee aplicaciones comerciales usadas alrededor del mundo, un alto grado de verificabilidad y una suave transición de sistemas de votación tradicionales a electrónicos, y mejorando así su accesibilidad y facilidad de uso.

V. Conclusiones

En este trabajo se ha definido un marco de evaluación, común para todos los sistemas de verificación de votación electrónica (SVVs), y se ha elaborado un estudio comparativo de una selección de dichos sistemas. Para ello se han seguido los siguientes pasos: (i) se ha definido una *clasificación* de SVVs, (ii) se han definido las *propiedades* a estudiar de los SVVs, (iii) se han *elegido y analizado* un conjunto representativo de SVVs tanto del mundo académico como comercial, y finalmente, (iv) se ha hecho un *estudio de tendencias*.

La clasificación de los SVVs se ha hecho combinando varios criterios con el propósito de conseguir una agrupación natural y completa adecuada a nuestro enfoque. No se consideran los sistemas de votación basados en papel. Se usa la clasificación HAVA [1] de EE.UU., que agrupa los SVVs en función del tipo de verificación. Se distingue también si el método de verificación es independiente del sistema de votación o si ha sido desarrollado para un sistema concreto de votación. A pesar de que existen muchas categorizaciones correctas, ésta ha sido la más adecuada para nuestros propósitos.

Cuadro III
Detalle de las propiedades de los SVVs.

Aspectos Técnicos	Flexibilidad		←	→	←	→	→	←
	Escalabilidad		→	→	≀	←	≀	←
	Disponibilidad		N/A	N/A	N/A	←	N/A	N/A
	Simplicidad		→	←	←	→	←	←
Integración	Gestión de la Información		N/A	NA/PD	N/A	A/NPD	N/A	A/NPD
	Integración		T/SW	T	SW	T/SW	N/A	N/A
Seguridad	**Relacionado con la Votación**	Auditabilidad	N	≀	N	Y	Y	Y
		Verificación Universal: Precisión Escrutinio	N	N	Y	Y	Y	Y
		Verificación Universal: Integridad de la Urna	≀	N	Y	Y	Y	Y
	Relacionado con el Votante	Verificación Individual	≀	N	Y	Y	Y	Y
		Resistencia Coacción	N	N	N	Y	Y	Y
		Anonimato	Y	N	Y	Y	Y	Y
		Secreto de Voto	N	N	Y	Y	Y	Y
Interacción con el Usuario	Confiabilidad		←	←	←	←	←	←
	Impacto de Uso		≀	←	←	→	→	←
	Accesibilidad		→	≀	→	←	N/A	←
	SVV		Frog	VVAATT VVVAT	VoteHere	Vote Box	Three-Ballot	E-valg

Las propiedades elegidas para evaluar los SVVs han sido la seguridad del sistema, la confianza del votante, y en menor medida la flexibilidad de voto. Aspectos como el secreto de voto, el anonimato del votante y la precisión de los resultados son sumamente importantes e imprescindibles en unas votaciones. No es menos importante la confianza que deben tener los votantes en el sistema de votación. En una democracia hace falta confiar en ella y en los mecanismos que la sustentan.

Este estudio se ha centrado en sistemas de verificación de elecciones electrónicas. Más concretamente, en los SVVs sin dependencia del papel, dado que los sistemas de votación han

evolucionado progresivamente hacia el uso de la tecnología digital y la criptografía. Teniendo en cuenta este subconjunto, se han elegido los sistemas más relevantes.

A partir de este estudio se puede concluir que los SVVs pretenden facilitar el voto a personas sordas mediante audio guías, así como aumentar la seguridad y la confianza del votante en el sistema. Estas últimas con la ayuda técnicas criptográficas, como son (i) el *cifrado* del voto, (ii) la *firma digital* y el *esquema umbral*, (iii) el *mixing* o los *homomorfismos*, (iv) las *ZKPs*, (v) los *códigos de retorno* y *recibos*, y (vi) la *auditoría* del sistema.

El cifrado del voto mediante RSA o ElGamal garantiza el secreto de voto. Con la firma digital y el esquema umbral se protege la autenticidad de los votos y se aumenta el secreto de voto. El mixing o las propiedades homomórficas derivadas de ElGamal sirven para conseguir el anonimato del votante. Este último tiene importantes consecuencias en el sistema, ya que aumenta la eficiencia del escrutinio [13], [14] y reduce la flexibilidad tanto en el tipo como en el formato del voto. En cambio, el uso del mixing proporciona un escrutinio ligeramente más lento [13], [14]. Ambas técnicas tienen sus ventajas y sus inconvenientes, pero las dos necesitan del uso de ZKPs para conseguir una verificación universal (también el anonimato en los que utilizan mixing). La gran mayoría de SVVs utilizan códigos de retorno y recibos para conseguir una verificación individual (E2E), evitar la coacción del votante y a su vez, aumentar la confianza del votante en el sistema. En la actualidad existe la tendencia de auditar, cada vez más, los sistemas informáticos de cualquier tipo por motivos de seguridad. Sólo los SVVs más modernos auditan todo lo sucedido a lo largo del proceso electoral.

Se puede observar cierta evolución en el uso del mixing y/o de homomorfismos. Los primeros sistemas homomórficos utilizaban las propiedades aditivas. En cambio, sistemas más contemporáneos emplean sus propiedades multiplicativas con el objetivo de aumentar la flexibilidad del voto. Otros sistemas llamados híbridos, aún más recientes, van más allá y combinan el mixing con la criptografía homomórfica multiplicativa para aprovechar las ventajas de ambos, y ganar en eficiencia y flexibilidad.

Los sistemas de votación electrónica actuales, como E-valg y VoteBox, hacen posible el voto *presencial remoto*. Esta posibilidad supone un nuevo paso hacia el voto plenamente *remoto*. De hecho, ya se han realizado experiencias de voto por *Internet*. Se argumenta que la implantación de los *sistemas remotos* facilitaría la realización de consultas, aumentando la participación ciudadana en la toma de decisiones. Por consiguiente acercaría un poco más la democracia a la sociedad. No obstante, la aceptación de estos esquemas dependerá del cumplimiento al mismo tiempo de dos requisitos: ofrecer un elevado nivel de *seguridad* y una gran facilidad de *acceso* a los votantes. En el apartado de la seguridad de la votación remota preocupa especialmente que el entorno de votación no esté supervisado, es decir, no se controla ni la plataforma del votante ni su entorno. Este hecho facilita la posibilidad de la coerción de los votantes y la venta masiva de votos.

AGRADECIMIENTOS

Los autores agradecen las ayudas del MICINN (proyectos eAEGIS TSI2007-65406-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004), del Ministerio de Industria, Comercio y Turismo (proyecto TSI-020100-2009-720), y del Govern de la Generalitat de Catalunya (ajuda 2009 SGR 1135). Los autores son responsables de las ideas expresadas en este artículo, que no reflejan necesariamente la posición de la UNESCO ni comprometen a dicha organización.

REFERENCIAS

[1] Election Assistance Commission (USA), "Voluntary voting system guidelines," 2005. Last visit: April 2010. [Online]. Available: http://www.eac.gov/voting%20systems/docs/vvsgvolumei.pdf/attachment_download/file

[2] A. T. Sherman, A. Gangopadhyay, S. H. Holden, G. Karabatis, A. G. Koru, C. M. Law, D. F. Norris, J. Pinkston, A. Sears, and D. Zhang, "An examination of vote verification technologies: findings and experiences from the Maryland study," in *Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop (EVT'06)*. Berkeley, CA, USA: USENIX Association, 2006, pp. 10–10.

[3] S. Bruck, D. Jefferson, and R. Rivest, "A modular voting architecture ("FROGS")," August 2001. [Online]. Available: http://www.vote.caltech.edu/drupal/files/working_paper/vtp_wp3.pdf

[4] T. Selker, "The voter verified audio audit transcript trail," September 2004. Last visit: February 2010. [Online]. Available: http://www.dos.state.pa.us/election_reform/lib/election_reform/VVAATT_CalTech.pdf

[5] E. Cross, G. Rogers, J. McClendon, W. Mitchell, K. Rouse, P. Gupta, P. Williams, I. Mkpong-Ruffin, Y. McMillian, E. Neely, J. Lane, H. Blunt, and J. Gilbert, "Prime III: One machine, one vote for everyone," in *Proceedings of 2007 Voting Competition Conference*, 2007, July.

[6] P. E. Varner, "Vote early, vote often, and VoteHere: A security analysis of VoteHere," Ph.D. dissertation, University of Virginia, March 2001.

[7] D. Sandler, K. Derr, and D. S. Wallach, "Votebox: a tamper-evident, verifiable electronic voting system," in *Proceedings of the 17th conference on Security symposium (SS'08)*. Berkeley, CA, USA: USENIX Association, 2008, pp. 349–364.

[8] A. O. Santin, R. G. Costa, and C. A. Maziero, "A Three-Ballot-based secure electronic voting system," *IEEE Security and Privacy*, vol. 6, no. 3, pp. 14–21, 2008.

[9] Norwegian Ministry of Local Gobernment and Regional Development, "E-vote 2011: Contractor solution specification," December 2009. Last visit: February 2010. [Online]. Available: http://www.regjeringen.no/upload/KRD/Kampanjer/valgportal/e-valg/e_valg_systemlosning/Tilbud_ergogroup/SSA-U_Appendix_2A_Contractor_Solution_Specification.pdf

[10] Norwegian Ministry of Local Gobernment and Regional Development, "E-vote 2011: Accessibility and usability evaluation of e-vote prototypes," November 2009. Last visit: February 2010. [Online]. Available: http://www.regjeringen.no/upload/KRD/Kampanjer/valgportal/e-valg/e_valg_systemlosning/report_evoting_usability_accessibility_eval_nr_iter2_final.pdf

[11] J. Benaloh, "Ballot casting assurance via voter-initiated poll station auditing," in *Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology (EVT'07)*. Berkeley, CA, USA: USENIX Association, 2007, pp. 14–14.

[12] R. Barnes, "VoteHere VHTi: A verifiable e-voting protocol," 2004. Last visit: April 2010. [Online]. Available: http://www.cs.virginia.edu/crab/VoteHere.pdf

[13] K. Peng, R. Aditya, C. Boyd, E. Dawson, and B. Lee, "Multiplicative homomorphic e-voting," in *Proceedings of 5th International Conference on Cryptology in India (INDOCRYPT 2004)*, 2004, pp. 61–72.

[14] K. Peng, "A hybrid e-voting scheme," in *Proceedings of the 5th International Conference on Information Security Practice and Experience (ISPEC '09)*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 195–206.

Sistema de peajes electrónicos seguro con anonimato revocable

Arnau Vives-Guasch*, Jordi Castellà-Roca*, Macià Mut-Puigserver† y Magdalena Payeras-Capella†
*Dept. de Ingeniería Informática y Matemáticas
Universitat Rovira i Virgili
Email: {arnau.vives, jordi.castella}@urv.cat
†Dept. de Ciencias Matemáticas e Informática
Universitat de les Illes Balears
Email: {macia.mut, mpayeras}@uib.es

***Resumen*—Los sistemas de peajes calculan la tarifa que el viajero debe pagar en función de los puntos de entrada y salida. Estos sistemas se basaban en el papel para realizar este control. La progresiva introducción de las tecnologías de la información y de las comunicaciones (TIC) permite la utilización de billetes electrónicos que reducen los costes y mejoran el control de las infraestructuras. No obstante, estos sistemas deben ser seguros frente a posibles fraudes, además de preservar la privacidad de sus usuarios. En este trabajo proponemos un sistema de peajes electrónicos que ofrece un elevado nivel de privacidad a los usuarios que actúan honestamente. El proveedor del servicio no conoce su identidad y no puede enlazar sus viajes. Sin embargo, si los usuarios actúan deshonestamente se puede revocar su anonimato.**

***Palabras clave*—peajes electrónicos, seguridad, privacidad, anonimato revocable, no-enlazabilidad, dispositivos móviles**

I. Introducción

La incorporación de las Tecnologías de la Información y las Comunicaciones (TIC) en los sistemas de peajes (en inglés Automatic Fare Collection systems - AFC) permite reducir costes y obtener mejoras de control en las infraestructuras, como podría ser la monitorización de la densidad del tráfico en tiempo real, o la planificación de las infraestructuras en función de los flujos de viajeros. Los sistemas AFC están diseñados para el transporte público masivo. El usuario no establece previamente su destino, sino que la tarifa se calcula en función del lugar por el que entra y del lugar por el que sale del sistema. En este sentido, es necesario habilitar una gestión segura de las entradas (*check-in*) y salidas (*check-out*), dado que los usuarios pagan en función de esta utilización. Si el sistema identifica a cada usuario y conoce sus movimientos, puede hacer un seguimiento de éstos vulnerando su privacidad. Por este motivo, los sistemas de peajes deben incorporar medidas para preservar la privacidad de los usuarios.

Nuestra propuesta ofrece una gestión segura del sistema de peajes y ofrece un elevado nivel de privacidad a los usuarios que actúan honestamente. No obstante, si un usuario no es honesto se revela su identidad de manera que puede ser castigado.

En primer lugar, en la sección II, se realiza un estado del arte en los sistemas de peajes (AFC) estudiando la seguridad y privacidad de las propuestas. Nuestra propuesta está descrita en la sección III y en la sección IV se realiza un breve análisis de seguridad. Finalmente, las conclusiones se presentan en la sección V.

II. Estado del Arte

Hemos realizado un estudio de las propuestas de AFC que tienen en cuenta el anonimato de los usuarios, y ofrecen anonimato revocable [8], [2], [3], [4], [5], [6] .

En la mayoría de ellas, el proveedor del servicio puede enlazar varios viajes de un mismo usuario [8], [2], [4], [5], [6]. En [3] el proveedor no puede enlazar los viajes de los usuarios, pero para ello los usuarios deben obtener una credencial diferente para cada viaje. Esto supone un sobrecoste importante en sistemas de transporte masivo, donde las entradas y las salidas deben ser lo más rápidas posibles.

En el caso de los dispositivos utilizados, observamos que la tendencia va en la línea de utilizar dispositivos móviles (p.ej. teléfonos móviles, PDAs, *smart phones*, etc.) [8], [3], [5], [6] para estos sistemas, imponiéndose a las tarjetas inteligentes (*smart-cards*) [2], [3], [4].

En el Cuadro I vemos de forma resumida la clasificación de las propuestas analizadas según el nivel de anonimato, la enlazabilidad, y el dispositivo que se utiliza en estos sistemas.

Ref.	Anonimato	Enlazable	Dispositivo
[8]	Revocable	Sí	Móvil
[2]	Revocable	Sí	Smart-card
[3]	Revocable	No	Móvil y Smart-card
[4]	Revocable	Sí	Smart-card
[5]	Revocable	Sí	Móvil
[6]	Revocable	Sí	Móvil

Cuadro I
Comparación de las propuestas analizadas

Nuestra propuesta ofrece anonimato revocable y no-enlazabilidad. La propuesta ha sido diseñada para que los usuarios utilicen sus dispositivos móviles en el sistema de peajes. Cabe destacar que los usuarios no necesitan obtener una credencial nueva cada vez que realizan un viaje, a diferencia de [3].

III. Sistema de peajes electrónicos

En esta sección describimos nuestro protocolo que protege el anonimato de los usuarios mediante firmas de grupo [1] para servicios de transporte masivo.

A continuación se describen los participantes del sistema, las propiedades de seguridad, la información de los billetes de entrada y salida, y las fases del sistema.

III-A. Participantes del sistema

En el sistema propuesto participan los siguientes actores:

- Usuario $\mathcal{U}$: accede al sistema de transporte y paga por el servicio recibido en la salida.
- Proveedor de servicios ($\mathcal{P}_\mathcal{S}$ es origen, $\mathcal{P}_\mathcal{D}$ es destino): estación que controla los billetes utilizados por $\mathcal{U}$.
- TTP de pago $\mathcal{M}_\mathcal{C}$: gestiona los pagos de los usuarios cuando salen del sistema.
- TTP de grupo $\mathcal{M}_\mathcal{G}$: gestiona las claves de grupo, las listas de revocación de los usuarios, etc. Tiene la potestad de revocar el anonimato de un usuario si éste actúa deshonestamente a partir de la firma de grupo de los billetes de entrada o salida.

III-B. Propiedades de seguridad

Los servicios de transporte siempre dan un recibo o billete a los usuarios para ser posteriormente validados, ejerciendo como prueba de que se ha actuado correctamente en el sistema. En estos sistemas, al ejecutarse totalmente de forma electrónica, se deben cumplir las siguientes propiedades de seguridad:

- Autenticidad: un billete debe ser generado por su correcto emisor.
- No-repudio: el emisor no puede negar la evidencia de haber generado uno de sus billetes.
- Integridad: el billete, una vez generado, no puede ser posteriormente modificado.

Además de estos requisitos básicos, aparecen otros que también deben cumplirse:

- Tiempo de validez: no puede utilizarse un billete una vez ha expirado su tiempo de validez.
- No-sobreutilización: no puede utilizarse un billete más veces que el número de usos preestablecido.
- Anonimato revocable: el sistema debe permitir el anonimato de los usuarios para obtener la aceptación de dicha comunidad, aunque el sistema y las autoridades públicas preferirían el no-anonimato para los usuarios por seguridad y control. De este modo, una solución viable y de compromiso es el anonimato revocable para los usuarios. Si un usuario actúa deshonestamente se revoca su anonimato.
- No-enlazabilidad: el proveedor únicamente debe poder enlazar la entrada con su correspondiente salida, evitando entonces la enlazabilidad entre varios viajes de un mismo usuario (*tracking*).

III-C. Información en los billetes

A continuación, mostramos la información que tienen los billetes de entrada (Cuadro II) y de salida (Cuadro III), además de mostrar la notación (Cuadro IV).

Billete de entrada ($\mathsf{t_{in}}^*$)		
Nombre	Notación	Descripción
Número de serie	Sn	generado por $\mathcal{P}_\mathcal{S}$
Estación de entrada	Ps	identificador de $\mathcal{P}_\mathcal{S}$
Timestamp de entrada	τ_1	marca de tiempo de entrada
Tiempo de validez	τ_v	tiempo para ser verificado
Commitment de $\mathcal{U}$	σ^*	compromiso del usuario firmado
Firma digital	$Sign_{\mathcal{P}_\mathcal{S}}(\mathsf{t_{in}})$	contenido firmado por $\mathcal{P}_\mathcal{S}$

Cuadro II
Información del billete de entrada

Billete de salida ($\mathsf{t_{out}}^*$)		
Nombre	Notación	Descripción
Información de validación	θ^*	enviado por $\mathcal{U}$
Tarifa	a	cantidad pagada
Timestamp de pago	τ_5	marca de tiempo del pago
Firma digital	$Sign_{\mathcal{P}_\mathcal{D}}(\mathsf{t_{out}})$	contenido firmado por $\mathcal{P}_\mathcal{D}$

Cuadro III
Información del billete de salida

Nombre	Notación
Clave pública de grupo	gpk
Lista de claves privadas para cada usuario del grupo	$gsk[\,]$
Lista de revocaciones del grupo	$grt[\,]$
Base de exponenciación	α
Número primo	p
Número primo	q
Seudónimo de $\mathcal{U}$ (para el pago)	$y_\mathcal{U}$
Exponenciación inversa de $y_\mathcal{U}$ (secreta)	$x_\mathcal{U}$
Número aleatorio	r
Exponenciación de r	δ_1
Encriptación probabilística de $y_\mathcal{U}$	δ_2
i-ésima marca de tiempo	τ_i
Firma digital del contenido $content$ por la entidad E	$Sign_E(content)$
Billete de entrada firmado por $\mathcal{P}_\mathcal{S}$	$\mathsf{t_{in}}^*$
Reto para $\mathcal{U}$ para demostrar autoría de $y_\mathcal{U}$	c
Reto y tarifa firmados por $\mathcal{P}_\mathcal{D}$ para $\mathcal{U}$	β^*
Respuesta de $\mathcal{U}$ al reto c	ω
Encriptación probabilística de ω	γ
Aceptación del cobro firmada por parte de $\mathcal{M}_\mathcal{C}$	ok^*
Billete de salida firmado por $\mathcal{P}_\mathcal{D}$	$\mathsf{t_{out}}^*$

Cuadro IV
Información de la notación, ordenada por orden de aparición

III-D. Fases del sistema

En el sistema se definen las fases siguientes:

- Configuración inicial: $\mathcal{M}_\mathcal{G}$ genera todas claves de grupo, listas de revocación, etc.
- Registro del usuario: $\mathcal{U}$ se registra en $\mathcal{M}_\mathcal{G}$, adquiriendo un par de claves de grupo. También crea una cuenta con $\mathcal{M}_\mathcal{C}$ mediante un seudónimo que será utilizado únicamente para los pagos.

- Entrada en el sistema: los usuarios entran en su estación origen y generan una firma de grupo que certifica que son usuarios válidos registrados del sistema. Esta firma no revela su identidad. A partir de esta firma reciben un billete de entrada que deben mostrar a la salida.
- Salida del sistema: el usuario se autentica otra vez en la estación de destino como usuario válido del grupo y muestra su billete de entrada. El proveedor $\mathcal{P}_\mathcal{D}$ calcula la cantidad que debe pagar el usuario a partir de la tarifa vigente. El usuario acepta el pago y genera la aceptación que se envía a $\mathcal{M}_\mathcal{C}$. A partir de la aceptación $\mathcal{M}_\mathcal{C}$ carga el importe a la cuenta de $\mathcal{U}$. Si todo el proceso es correcto, el usuario recibe un billete de salida.

III-E. Configuración inicial

Esta fase se ejecuta únicamente una vez para el conjunto de usuarios. $\mathcal{M}_\mathcal{G}$ genera el grupo de tamaño establecido, generando como salida $(gpk, gsk[\,], grt[\,], \alpha, p, q)$, siendo gpk la clave pública compartida del grupo, cada clave privada del usuario es $gsk[i]$, la lista de usuarios revocados en el grupo es $grt[\,]$, y (α, p, q) son parámetros públicos, siendo α la base pública de exponenciación, y (p, q) números primos tales que $p = 2q + 1$, cardinales de sus grupos correspondientes $\mathbb{Z}_p$ y $\mathbb{Z}_q$. Además, los proveedores de servicio generarán sus propias parejas de claves mostrando sus respectivas claves públicas. Las claves privadas de los usuarios $gsk[i]$ serán entregadas en el momento del registro de cada usuario.

III-F. Registro del usuario

$\mathcal{U}$ se registra en la TTP de grupo $\mathcal{M}_\mathcal{G}$ y recibe la pareja de claves de grupo $(gpk, gsk[i])$. A continuación, $\mathcal{U}$ también se registra en la TTP de pago $\mathcal{M}_\mathcal{C}$; el usuario tiene un seudónimo siendo una exponenciación precalculada $y_\mathcal{U} = \alpha^{x_\mathcal{U}} \pmod p$ dado un cierto valor aleatorio $x_\mathcal{U} \overset{R}{\leftarrow} \mathbb{Z}_q$; únicamente la información $y_\mathcal{U}$ será mostrada a la TTP de pago $\mathcal{M}_\mathcal{C}$, y autenticada mediante Schnorr [7] demostrando que se conoce $x_\mathcal{U}$ sin darlo a conocer. De este modo, se preserva el anonimato para el usuario, pero podría ser revocado por $\mathcal{M}_\mathcal{G}$ si fuera necesario.

III-G. Entrada en el sistema

Cuando el usuario $\mathcal{U}$ entra correctamente en el sistema recibe un billete de entrada $\mathsf{t_{in}}$. A la salida del sistema $\mathcal{U}$ debe mostrar el billete para calcular la cantidad que debe pagar. A continuación se describe el protocolo de entrada.

1. El usuario $\mathcal{U}$ realiza los pasos siguientes:
   a) genera un valor aleatorio $r \overset{R}{\leftarrow} \mathbb{Z}_q$;
   b) calcula $\delta_1 = \alpha^r \pmod p$;
   c) calcula $\delta_2 = PK_{\mathcal{M}_\mathcal{C}}(y_\mathcal{U})$ (el criptosistema utilizado es probabilístico);
   d) genera un *timestamp* τ_0;
   e) compone $\sigma = (\delta_1, \delta_2, \tau_0)$, y lo firma con su clave de grupo $\sigma^* = (\sigma, Sign_\mathcal{G}(\sigma))$;
   f) envía σ^* a $\mathcal{P}_\mathcal{S}$;
2. El proveedor de servicios origen $\mathcal{P}_\mathcal{S}$ realiza los pasos siguientes:
   a) verifica la firma de σ^*, es decir, comprueba si es un usuario válido del grupo y que no haya sido revocado anteriormente;
   b) genera un *timestamp* τ_1;
   c) rellena la información del billete de entrada al sistema $\mathsf{t_{in}} = (\mathsf{Sn}, \mathsf{Ps}, \tau_1, \tau_v, \sigma^*)$ y calcula la firma $\mathsf{t_{in}}^* = (\mathsf{t_{in}}, Sign_{\mathcal{P}_\mathcal{S}}(\mathsf{t_{in}}))$;
   d) envía $\mathsf{t_{in}}^*$ a $\mathcal{U}$;
3. $\mathcal{U}$ verifica la firma de $\mathsf{t_{in}}^*$ y su contenido;

III-H. Salida del sistema

Cuando el usuario sale del sistema, envía el billete de entrada $\mathsf{t_{in}}$ a la estación de salida $\mathcal{P}_\mathcal{D}$, y se calcula la cantidad que debe pagar. Si $\mathcal{U}$ actúa honestamente recibe un billete de salida $\mathsf{t_{out}}$ como recibo del pago.

1. $\mathcal{U}$ envía $\mathsf{t_{in}}^*$ a $\mathcal{P}_\mathcal{D}$;
2. El proveedor de servicios destino $\mathcal{P}_\mathcal{D}$ realiza los pasos siguientes:
   a) verifica la firma de $\mathsf{t_{in}}^*$ calculada por $\mathcal{P}_\mathcal{S}$;
   b) comprueba que $\mathsf{t_{in}.Sn}$ no haya sido utilizado en anterioridad;
   c) verifica que no haya expirado el tiempo de validez τ_v;
   d) obtiene un *timestamp* τ_2;
   e) calcula la cantidad a pagar dependiendo del punto de entrada ($\mathsf{t_{in}.Ps}$), de salida (Pd) y de sus respectivos tiempos ($\mathsf{t_{in}}.\tau_1$ y τ_2): $a = f(\mathsf{t_{in}.Ps}, \mathsf{Pd}, \mathsf{t_{in}}.\tau_1, \tau_2)$;
   f) genera un reto $c \overset{R}{\leftarrow} \mathbb{Z}_q$;
   g) compone $\beta = (\mathsf{t_{in}}^*, a, c, \tau_2, \mathsf{Pd})$, y lo firma $\beta^* = (\beta, Sign_{\mathcal{P}_\mathcal{D}}(\beta))$;
   h) envía β^* a $\mathcal{U}$;
3. $\mathcal{U}$ realiza los pasos siguientes:
   a) verifica la firma de β^* calculada por $\mathcal{P}_\mathcal{D}$;
   b) calcula $\omega = r + c \cdot x_\mathcal{U} \pmod q$;
   c) genera un *timestamp* τ_3;
   d) compone y cifra $\gamma = PK_{\mathcal{M}_\mathcal{C}}(\omega, \mathsf{t_{in}.Sn}, \tau_3, a)$;
   e) compone $\theta = (\beta^*, \gamma, \tau_3)$ y lo firma con la firma de grupo: $\theta^* = (\theta, Sign_\mathcal{G}(\theta))$;
   f) envía θ^* a $\mathcal{P}_\mathcal{D}$;
4. $\mathcal{P}_\mathcal{D}$ verifica la firma de θ^* y su contenido, y la envía a la TTP de pago $\mathcal{M}_\mathcal{C}$;
5. $\mathcal{M}_\mathcal{C}$ realiza los pasos siguientes:
   a) verifica la firma de θ^*;
   b) descifra γ para obtener la prueba de Schnorr para ser verificada ω;
   c) descifra $\sigma.\delta_2$ para obtener el seudónimo a quién cargar la cuenta del viaje $y_\mathcal{U}$;
   d) verifica la identidad de $\mathcal{U}$ mediante Schnorr: $\alpha^\omega \overset{?}{=} \delta_1 \cdot (y_\mathcal{U})^c$;
   e) si es correcto, carga la cuenta del viaje con importe a al usuario que apunta $y_\mathcal{U}$;
   f) genera un *timestamp* τ_4;
   g) genera $ok = (\mathsf{t_{in}.Sn}, a, \tau_4)$ y lo firma $ok^* = (ok, Sign_{\mathcal{M}_\mathcal{C}}(ok))$;

h) envía ok^* a $\mathcal{P}_\mathcal{D}$;

6. $\mathcal{P}_\mathcal{D}$ realiza los pasos siguientes:
   a) genera un *timestamp* τ_5;
   b) compone $\mathsf{t_{out}} = (\theta^*, a, \tau_5)$ y lo firma $\mathsf{t_{out}}^* = (\mathsf{t_{out}}, Sign_{\mathcal{P}_\mathcal{D}}(\mathsf{t_{out}}))$;
   c) envía $\mathsf{t_{out}}^*$ a $\mathcal{U}$ y permite salir al usuario del sistema;

IV. Análisis de seguridad

Proposición IV.1. *El sistema propuesto preserva la autenticidad, el no-repudio y la integridad de los billetes de entrada y salida.*

Afirmación 1. No es posible la creación de billetes de entrada o salida fraudulentos.

Prueba. Los billetes tienen la forma siguiente $\mathsf{t_{in}}^* = (\mathsf{t_{in}}, Sign_{\mathcal{P}_\mathcal{S}}(\mathsf{t_{in}}))$ y $\mathsf{t_{out}}^* = (\mathsf{t_{out}}, Sign_{\mathcal{P}_\mathcal{D}}(\mathsf{t_{out}}))$, además de la información enviada previa al pago $\beta^* = (\beta, Sign_{\mathcal{P}_\mathcal{D}}(\beta))$. Si una entidad no autorizada puede crear un billete (entrada o salida) válido sin disponer de las claves privadas de $\mathcal{P}_\mathcal{S}$ ni de $\mathcal{P}_\mathcal{D}$, podría generar firmas digitales en nombre de los proveedores. Suponiendo la utilización de un esquema de firma digital seguro, esta operación no se considera posible. Por otro lado, el usuario envía la información de verificación firmada con su clave de grupo: $\sigma^* = (\sigma, Sign_\mathcal{G}(\sigma))$ y $\theta^* = (\theta, Sign_\mathcal{G}(\theta))$. Por el motivo anterior, esta firma garantiza que el mensaje es auténtico y ha sido enviado por un usuario válido (no revocado) dentro del grupo.

Afirmación 2. El emisor de un billete no puede denegar la emisión de dicho billete.

Prueba. Los billetes están firmados por su emisor (los proveedores de servicio) y, considerando que el esquema de firma es seguro, esta operación solamente la pueden realizar ellos. Por lo tanto, la identidad del emisor está asociada al billete y por las propiedades del esquema de firma electrónica no pueden negar su autoría. De esta misma manera ocurre con la firma de grupo, donde si la identidad se revela, puede verificarse la autoría de un mensaje.

Afirmación 3. El contenido de un billete no puede ser modificado.

Prueba. Suponiendo que el esquema de firma es seguro y que la función resumen utilizada en la firma es resistente a colisiones, si se modifica el contenido del billete la verificación de la firma de los billetes será incorrecta. Para que la verificación fuera correcta se debería volver a generar la firma realizada sobre el resumen del nuevo contenido. Como se ha mencionado anteriormente, esta operación no es posible con los equipos actuales. El mismo argumento se puede aplicar con la firma de grupo.

Resultado IV.1. *De acuerdo con las definiciones en la sección III-B y las Afirmaciones 1, 2 y 3, podemos asegurar que el protocolo consigue las propiedades de autenticidad, no-repudio e integridad.*

Proposición IV.2. *El sistema descrito en la sección III satisface la propiedad de anonimato revocable, y los movimientos de un mismo usuario no son enlazables entre sí por parte de los proveedores.*

Afirmación 4. Un billete es anónimo.

Prueba. La información relativa a la identidad del usuario está cifrada con la clave pública de la TTP de pago. Los proveedores ($\mathcal{P}_\mathcal{S}$ y $\mathcal{P}_\mathcal{D}$) no pueden acceder a esta información al no disponer de la clave privada de la TTP. En el sistema los usuarios calculan dos firmas de grupo ($\mathsf{t_{in}}.\sigma^* = (\sigma, Sign_\mathcal{G}(\sigma))$ y $\theta^* = (\theta, Sign_\mathcal{G}(\theta))$) que certifican que el firmante es un usuario válido del grupo. Por las propiedades de las firmas de grupo los proveedores no pueden obtener la identidad del emisor de la firma. Si hubiese algun problema, podría obtenerse la identidad del usuario que generó la firma mediante la cooperación de las distintas TTP de pago $\mathcal{M}_\mathcal{C}$ y de grupo $\mathcal{M}_\mathcal{G}$. Si el usuario aparece en la lista de revocación, se descubre su identidad, permitiendo entonces las acciones pertinentes.

Afirmación 5. El usuario es anónimo frente a los proveedores en la operación de pago.

Prueba. Toda la información relacionada con el pago está cifrada y sólo la TTP de pago puede acceder a ella. Las estaciones son externas al pago, y sólo reciben la confirmación de parte de la TTP de pago que éste se ha realizado correctamente. La TTP de pago $\mathcal{M}_\mathcal{C}$ conoce $y_\mathcal{U}$ de la dupla $(x_\mathcal{U}, y_\mathcal{U})$ donde $y_\mathcal{U} = \alpha^{x_\mathcal{U}} \pmod p$ que lo identifica como usuario; entonces, el usuario se autentica mostrando el conocimiento de su pareja $x_\mathcal{U}$ mediante la autenticación de Schnorr [7].

Afirmación 6. Múltiples firmas de grupo realizadas por un mismo usuario no son enlazables entre sí, por parte de los proveedores.

Prueba. La propuesta de firma de grupo realizada por Boneh y Shacham [1] utiliza un modelo de firma probabilístico, es decir, no es posible predecir un texto cifrado dado un texto plano como entrada. Ello permite la no-enlazabilidad entre diferentes firmas de grupo realizadas por un mismo usuario.

Resultado IV.2. *De acuerdo con las definiciones dadas en la sección III-B y las Afirmaciones 4, 5 y 6, podemos asegurar que el protocolo consigue las propiedades de anonimato revocable y no-enlazabilidad.*

Proposición IV.3. *El protocolo no permite la sobreutilización de los billetes, además de garantizar el cumplimiento de sus fechas de vencimiento.*

Afirmación 7. El protocolo no permite la sobreutilización de los billetes.

Prueba. Si un usuario intenta sobreutilizar un billete (de entrada), quedará patente el número de serie utilizado con anterioridad. Si se demuestra este acto deshonesto por parte del usuario, se puede recurrir a la TTP de grupo $\mathcal{M}_\mathcal{G}$ para que lo incluya en la lista de revocación.

Afirmación 8. El billete no puede ser válido después de la expiración de su tiempo de validez τ_v.

Prueba. La estación de salida $\mathcal{P}_{\mathcal{D}}$ recibe el billete del usuario para ser verificado. En esta verificación, se comprueba que el tiempo actual no haya sobrepasado el tiempo de validez τ_v explícito en el mismo billete $\mathsf{t_{in}}^*$ firmado por $\mathcal{P}_{\mathcal{S}}$.

Resultado IV.3. *De acuerdo con las definiciones dadas en la sección III-B y las Afirmaciones 7 y 8, podemos asegurar que el protocolo consigue las propiedades de no-sobreutilización y el cumplimiento de la fecha de vencimiento.*

V. Conclusiones

Se ha realizado una propuesta de peajes electrónicos adaptada a los servicios de transporte masivo de usuarios, utilizando firmas de grupo para preservar su privacidad. Las firmas de grupo permiten revocar el anonimato en el caso de actuación corrupta de algún usuario. Nuestra propuesta a diferencia de las anteriores no requiere que los usuarios obtengan una credencial nueva en cada viaje para conseguir la no-enlazabilidad.

A partir de esta propuesta, la línea a seguir va en la dirección de extender el protocolo implementando un demostrador para dispositivos móviles, aprovechando las ventajas de reducción de recursos (más rapidez y claves más cortas) que nos permiten los operadores bilineales en los que trabaja la firma de grupo de Boneh y Shacham [1] a través de criptografía de curva elíptica.

Agradecimientos

Los autores agradecen las ayudas del MICINN (proyectos eAEGIS TSI2007-65406-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004, TSI2007-62986, "RIPUP" TIN2009-11689), del Ministerio de Industria, Comercio y Turismo (proyecto eVerification TSI-020100-2009-720), y del Gobierno de Catalunya (ayuda 2009 SGR 1135). Los autores son responsables de las ideas expresadas en este artículo, que no reflejan necesariamente la posición de la UNESCO ni comprometen a dicha organización.

Referencias

[1] D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, pages 168–177, New York, NY, USA, 2004. ACM.

[2] L. Buttyán, T. Holczer, and I. Vajda. Providing location privacy in automated fare collection systems. In *In Proceedings of the 15th IST Mobile and Wireless Communication Summit, Mykonos, Greece*, June 2006.

[3] T. S. Heydt-Benjamin, H.-J. Chae, B. Defend, and K. Fu. Privacy for public transportation. In *6th Workshop on Privacy Enhancing Technologies (PET 2006)*, pages 1–19, 2006. LNCS 4258.

[4] S.-P. Hong and S. Kang. Ensuring privacy in smartcard-based payment systems: A case study of public metro transit systems. In *Communications and Multimedia Security*, pages 206–215, 2006.

[5] O. Jorns, O. Jung, and G. Quirchmayr. A privacy enhancing service architecture for ticket-based mobile applications. In *2nd International Conference on Availability, Reliability and Security*, pages 374–383, Vienna, Austria, Apr 2007. ARES 2007 - The International Dependability Conference. vol. 24.

[6] G. Madlmayr, P. Kleebauer, J. Langer, and J. Scharinger. Secure communication between web browsers and nfc targets by the example of an e-ticketing system. In *EC-Web '08: Proceedings of the 9th international conference on E-Commerce and Web Technologies*, pages 1–10, Berlin, Heidelberg, 2008. Springer-Verlag.

[7] C.-P. Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, 1991.

[8] H. Wang, J. Cao, and Y. Zhang. Ticket-based service access scheme for mobile users. *Aust. Comput. Sci. Commun.*, 24(1):285–292, 2002.

Sobre la comparación de mensajes cifrados y su aplicación en una red de sensores inalámbrica

Vanesa Daza
Departament de Tecnologies de la Informació i les Comunicacions
Universitat Pompeu Fabra
Email: vanesa.daza@upf.edu

***Abstract*—En este artículo estudiamos el caso particular de la comparación de dos valores a partir de sus correspondientes cifrados. Más concretamente, proponemos un protocolo en el que dados los cifrados de dos valores m_0 y m_1, se obtiene como resultado el cifrado del valor mayor, esto es un cifrado de $\max\{m_0, m_1\}$, sin que se revele el valor de m_0 y m_1. Para ello utilizamos como primitiva los homomorfismos de privacidad aditivos y multiplicativos.**

Nos hemos centrado en un escenario en el que un conjunto de sensores forman una red de sensores inalámbrica. Utilizaremos para ello el homomorfismo de privacidad de Domingo-Ferrer, fuertemente utilizado en encenarios de redes de sensores. Nuestra propuesta representa una reducción logarítmica en la medida del mensaje respecto a las propuestas existentes.

I. Introduction

Posiblemente debido a las múltiples aplicaciones que han surgido en el mundo real, en los últimos años la investigación en el campo de las redes de sensores inalámbricas ha sido muy intensa. Desde redes capaces de monitorizar la posibilidad de un incendio forestal en un bosque hasta redes que facilitan el aparcamiento en las grandes ciudades, mejorando de esta manera en gran medida el tráfico, múltiples aplicaciones han surgido con el principal objetivo de mejorar la calidad de vida.

Un problema común en las redes de sensores inalámbricas es gestionar la información que envían los sensores de la red en respuesta de una petición realizada por una baliza o sensor especial. El cifrado de cada uno de los mensajes que envía un sensor así como el descifrado de todos los mensajes recibidos es una de las primeras propuestas, pero resulta altamente costosa para los sensores así como poco segura. Es deseable, pues, evitar tener que descifrar los mensajes a cada paso. Por ello, otra de las propuestas que se han hecho ha sido añadir cada uno de los mensajes a medida que un sensor añadía sus datos. El principal problema de esta propuesta es que la longitud del mensaje es muy elevada (en función del número de sensores, que acostumbra a ser muy alto). Así, se introdujeron los homomorfismos de privacidad para la agregación de datos cultos (CDA, del inglés, Concealed Data Aggregation) [7], [8], que permitían agregar los mensajes cifrados obteniendo los resultados deseados en los mensajes en claro. Diferentes propuestas se encuentran en la literatura. Por agregar, entendemos aquí que se calcula una cierta función prefijada en un principio y cuyos argumentos son los valores que aportan cada uno de los sensores. Generalmente, esta función es simplemente la suma de los valores, aunque en ocasiones, otras funciones más sofisticadas como aquellas que proporcionan patrones de detección de movimiento son necesarias [7], dependiendo de los diferentes escenarios de la red de sensores en consideración.

Una función sencilla y enormemente útil en el cálculo distribuido es la función que permite calcular la comparación de sus argumentos. De hecho, no sólo es útil en escenarios relacionados con redes de sensores, si no que también es deseable en numerosos escenarios más generales. En ella se basan operaciones como el cálculo de valores máximos o mínimos. Encontrar estos valores puede resultar especialmente útil en aplicaciones donde los sensores toman medidas en un entorno concreto, ya sea, por ejemplo, un sistema de monitorización de incendios o sensores de temperatura o presión. Sin embargo, no se conocen muchas soluciones y mucho menos que sean eficientes. En [2] se propone una comparación segura de datos cifrados basándose en una esquema particular de cifrado que mantiene el orden [3], pero la propuesta no es eficiente. Posteriormente, en [6], los autores proponen una alternativa también basada en los homomorfismos de privacidad, pero la longitud de los mensajes es extremadamente grande (función de la longitud del espacio de los mensajes en claro), por lo que no es conveniente en escenarios como redes de sensores inalámbricas.

A. Contribución

En este artículo nos centramos en el caso particular de la comparación de dos valores a partir de sus cifrados. Más concretamente, proponemos un protocolo en el que dados los cifrados de dos valores m_0 y m_1, se obtiene el cifrado del $\max\{m_0, m_1\}$ sin necesidad de descifrar cada uno de los valores para proceder a la comparación. No se obtiene información de los valores m_0 ni m_1 al final del protocolo, a parte de la que se pueda deducir del propio resultado.

Para ello hacemos uso de las propiedades de un homomorfismo de privacidad aditivo y multiplicativo. Pese a que tiene interés en entornos más generales, nos hemos centrado en el escenario de las redes de sensores, donde es conocida la necesidad de la comparación de valores. Por ello, a modo de ejemplo, describimos el protocolo utilizado el homomorfismo de privacidad de Domingo-Ferrer, ampliamente utilizado en escenarios de redes de sensores. Para otras aplicaciones, sólo es necesario considerar un homomorfismo de privacidad que se adecúe al escenario. Nuestra propuesta representa una

reducción logarítmica en la medida del mensaje respecto a previas propuestas.

B. Organización del artículo

El resto del artículo está organizado de la siguiente manera. En la Sección II revisamos el concepto de homomorfismo de privacidad, detallando en concreto el funcionamiento del homomorfismo de privacidad propuesto por Domingo-Ferrer en [5]. En la Sección III introducimos el protocolo que nos permitirá a dos sensores comparar el texto en claro de dos mensajes cifrados sin necesidad de descifrarlos. Finalmente, concluiremos en la Sección IV, incluyendo algunas líneas futuras de investigación relacionadas.

II. Homomorfismos de Privacidad

El concepto de homomorfismo de privacidad (PH, del inglés Privacy Homomorphism) fue introducido por Rivest *et al.* en el año 1978 [11]. Un homomorfismo de privacidad es una transformación de cifrados que permite el cálculo de ciertas operaciones en los respectivos mensajes sin cifrar. Las transformaciones más usadas son la suma y el producto, dando lugar a los homomorfismos de privacidad aditivos y multiplicativos, respectivamente. Así, si llamamos C a la función de cifrado, D a la de descifrado, y $\otimes$ es una operación matemática relacionada con el propio homomorfismo en consideración (en general, será la suma y la multiplicación) el homomorfismo de privacidad aditivo permitiría calcular

$$a + b = D(C(a) \otimes C(b))$$

mientras que el multiplicativo permitiría el cálculo de

$$a \times b = D(C(a) \otimes C(b)).$$

Existen numerosas propuestas de homomorfismos de privacidad en la literatura, tanto de clave simétrica como asimétrica. Entre otros, podemos citar [11], [9] o [10]. Algunos propuestas, como [5], [1], [4] son tanto aditivas como multiplicativas.

Puesto que haremos un uso extensivo del homomorfismo de Domingo-Ferrer [5] en las siguientes secciones, pasamos a describirlo con detalle a continuación.

A. Esquema de Domingo-Ferrer

El homomorfismo de privacidad de Domingo-Ferrer es simétrico, por lo que se utiliza la misma clave para cifrar un mensaje que para descifrarlo. Es probabilístico, ya que en el proceso de cifrado se utilizan ciertos valores aleatorios que permiten que cifrados de un mismo mensaje sean diferentes.

El esquema considera un entero positivo $d \geq 2$ y un entero m que debe tener muchos divisores pequeños así como muchos enteros menores que m que puedan ser invertidos módulo m. Estos parámetros son públicos.

La clave secreta es $k = (r, m')$, donde $r \in \mathbb{Z}_m$ y $r^{-1} \mod m$ existe. El valor $s = \log_{m'} m$ es un parámetro de seguridad.

Sea $a \in \mathbb{Z}_{m'}$ el valor que se quiere cifrar. El cifrado y descifrado serían como sigue:

- Cifrado: Se divide el valor $a \in \mathbb{Z}_m$ aleatóriamente en d fragmentos $a_1, \ldots, a_d$, de manera que $a = \sum_{i=1}^{d} a_i \mod m'$ i $a_i \in \mathbb{Z}_m$ y se calcula:

$$E_k(a) = (a_1 r \mod m, \ldots, a_d r^d \mod m).$$

- Descifrado: Calculando, r^{-j} se obtiene la j-ésima coordenada, esto es, a_j. Después de efectuar este paso para todas las coordenadas, se calcula:

$$D_k(E_k(a)) = \sum_{i=1}^{d} a_j \mod m'.$$

Como hemos comentado anteriormente, este homomorfismo es aditivo y multiplicativo. Para calcular la suma de los mensajes sin cifrar a partir de los propios cifrados se debe calcular la suma, componente a componente, de los cifrados. Para calcular el producto de los mensajes sin cifrar funciona de manera semejante al producto de polinomios, interpretando la d tupla como los coeficientes de un polinomio de grado $d-1$. Se calcula el producto cruzado de todos los términos, esto es, los de grado d_1 con los de grado d_2, resultando en un término de grado $d_1 + d_2$. Finalmente, los que tienen igual grado se suman.

B. Cómo Agregar Datos con Homomorfismos de Privacidad

Los homomorfismos de privacidad han sido muy utilizados para agregar datos que se encuentran cifrados proporcionando así confidencialidad a las redes de sensores inalámbricas de manera eficiente. Así, cuando un sensor recibe información cifrada de otro nodo de la red, agrega su valor a través del cifrado correspondiente, y reenvía el resultado al siguiente nodo sin necesidad en ningún momento de descifrar el mensaje que se ha recibido. Por agregar entendemos aquí al cálculo de una cierta función prefijada en un principio y cuyos argumentos son los valores que aportan cada uno de los sensores. Generalmente, esta función es simplemente la suma de los valores, aunque en ocasiones otras funciones más sofisticadas como aquellas que proporcionan patrones de detección de movimiento son necesarias, dependiendo de los diferentes escenarios de la red de sensores en consideración.

Aquellos nodos de la red que agregan datos se conocen comúnmente como agregadores. Dependiendo de la naturaleza de la propia red, se asume que sólo algunos nodos tienen las características suficientes para agregar los datos o bien cualquier nodo es capaz de realizar dichas operaciones.

Como hemos comentado anteriormente, aunque existen homomorfismos de privacidad basados en el paradigma de la clave pública, la complejidad de los algoritmos así como las limitaciones de los sensores, hacen que en general no sea la opción más utilizada para garantizar la confidencialidad de los datos que circulan por la red. Así, uno de los homomorfismos más utilizados es el de Domingo-Ferrer, de clave privada, y que hemos descrito en la Sección II-A.

Aunque en [12], Wagner mostró una ataque con texto en claro escogido del homomorfismo que hemos descrito en II-A para ciertos parámetros, como se muestra en [13], el

homomorfismo de privacidad de Domingo-Ferrer resulta más seguro que cifrar y descifrar en cada uno de los sensores. Existen propuestas más seguras, especialmente considerando homomorfismos en el paradigma de clave pública, sin embargo el coste computacional y de comunicaciones hace que tales soluciones en general no sean deseables en escenarios de redes de sensores, especialmente en protocolos que hacen un uso intensivo de estos homomorfismos. De hecho, los autores también apuntan que para un adversario que intente obtener información confidencial, es razonable romper el mecanismo siempre y cuando el coste de romper el esquema sea mejor que el valor de la información que se cifra.

III. Comparación de valores cifrados

En esta sección describiremos el protocolo que nos permite comparar dos valores a partir de sus respectivos cifrados obteniendo como resultado el cifrado del valor que resulta mayor en la comparación. El protocolo es válido considerando cualquier homomorfismo de privacidad que permita tanto la suma como la multiplicación. Por ceñirnos a un escenario con una red de sensores inalámbrica, consideraremos el homomorfismo de privacidad de Domingo-Ferrer, ampliamente utilizado en entornos de redes de sensores, ya que resulta especialmente eficiente en estos entornos.

Supongamos que un sensor S_1 recibe un valor cifrado $C_k(m_0)$ de otro sensor S_0 de la red, donde C_k es un homomorfismo de privacidad simétrico como el descrito en la Sección II-A. El mensaje m_0 pertenece a un rango $[-R, R]$, donde R es un valor conocido *a priori*. Supongamos que el sensor S_1 tiene el valor $m_1 \in [-R, R]$. Evidentemente, puede calcular $C_k(m_1)$. A continuación describiremos un protocolo que nos permitirá calcular $C_k(f(m_0, m_1))$, donde $f(m_0, m_1) = \max\{m_0, m_1\}$ sin necesidad de descifrar en ningún momento los mensajes, solución que no resulta deseable por la complejidad de cálculo que puede acarrear y especialmente porque en los momentos en los que los datos se encuentran descifrados se encuentran especialmente vulnerables a los ataques de intrusos. Observermos que en este caso, el sensor S_1 podrá deducir si m_0 es mayor o menor que m_1, pero no cuál es su valor exacto. Esto es debido a que en este caso el sensor S_1 parte conociendo el valor de m_1. En el caso general, donde sólo se conocer los valores cifrados tal deducción no es posible.

A. Escenario

Sea $\mathcal{S}$ una red de sensores y sean $S_1, \ldots, S_\ell$ los sensores que integran la red. Supondremos que una baliza o sensor fuente (incluso podemos asumir que se trata del propietario de la red) lanza a los sensores de la red una determinada petición, como puede ser, por ejemplo, que cada sensor mida la temperatura ambiental del lugar donde se encuentre ubicado. El mensaje lo recogen un grupo de sensores que se encuentran en una zona físicamente más próxima a la baliza y lo dispersan a través del resto de sensores. Así, cada sensor S_i tendrá un valor m_i, para todo $i \in \{1, \ldots, \ell\}$. Supondremos que cada sensor conoce una clave secreta $k = (r, m')$, según el criptosistema de Domingo-Ferrer descrito en la Sección II-A. Se requiere además que el valor m' sea suficientemente grande de manera que el rango de posibles valores que puedan obtener cada uno de los valores de la red sea menor que m', y así sea posible comparar los valores heredando el orden natural de los enteros.

Sería deseable disponer de un dispositivo que garantice el uso de los valores secretos para poder cifrar sin que sea posible (o al menos, sea costoso) acceder a ellos. Ello prevendría de posibles ataques de intrusos. Notaremos $C_k(\cdot)$ al cifrado de un mensaje utilizando este homomorfimos y $D_k(\cdot)$ al proceso de descifrado, y definiremos como $C_k^r(\cdot)$, para $r \in \mathbb{N}$, la aplicación que cifra utilizando $C_k(\cdot)$ en cada una de las r coordenadas de la r-tupla, esto es, $C_k^r(\cdot) = C_k(\cdot) \times \ldots \times C_k(\cdot)$. Por simetría definimos de la misma manera $D_k^r(\cdot) = D_k(\cdot) \times \ldots \times D_k(\cdot)$.

Nos centraremos en el caso de la comparación de dos mensajes cifrados. Esto es, supondremos, como ya hemos dicho anteriormente, que un sensor S_1 recibe un valor cifrado $C_0 = C_k(m_0)$ de otro sensor S_0 de la red y que el objetivo es comparar el mensaje m_0 del sensor S_0 con el mensaje m_1 que tiene el sensor S_1 sin necesidad de descifrar C_0. Estamos suponiendo pues que todos los sensores pueden y deben efectuar la comparación. En el caso de que no sea así, y no todos los sensores sean agregadores, si no sólo algunos de ellos, el protocolo que describiremos a continuación en la Sección III-B lo llevará a cabo el sensor agregador con la diferencia de que no habrá recibido un único mensaje cifrado, si no un conjunto de r de ellos, por lo que deberá aplicar reiteradamente y de manera óptima nuestro protocolo para obtener el resultado deseado, utilizando cualquier protocolo que permita comparar de manera eficiente los diferentes valores del conjunto.

B. Protocolo

Dados los sensores S_0 y S_1, con sendos mensajes m_0 y m_1, el protocolo que proponemos nos permitirá obtener un cifrado de $\max\{m_0, m_1\}$ sin necesidad de recuperar ni m_0 ni m_1.

Para ello, el sensor S_0 debe enviar a S_1 el texto cifrado de la siguiente manera. El sensor S_0 calcula la representación binaria de m_0, que notaremos $(m_0)_2 = (\alpha_r, \alpha_{r-1}, \ldots, \alpha_1)$, donde $r > K$, siendo $K = \log_2 R$. Completaremos con zeros por la izquierda hasta completar la tupla de longitud r. A continuación, calcula $C_0 = C_k^r((m_0)_2) = (C_k(\alpha_r), \ldots, C_k(\alpha_1))$.

Una vez S_1 recibe C_0, calcula $(m_1)_2 = (\beta_r, \beta_{r-1}, \ldots, \beta_1)$, y $C_1 = C_k^r((m_1)_2) = (C_k(\beta_r), \ldots, C_k(\beta_1))$.

A continuación, S_1 calcula $C_{1,0} = C_1 - C_0$ y $C_{1,0}^\star = C_{1,0} \star C_{1,0}$, donde $\star$ representa el producto componente a componente de los dos vectores. Debido a las propiedades homomórficas del criptosistema utilizado, observemos que $C_{1,0}^\star = (C_k(\gamma_r), \ldots, C_k(\gamma_1))$, donde

$$\gamma_i = \begin{cases} 0, & \text{si } \alpha_i = \beta_i \\ 1, & \text{si } \alpha_i \neq \beta_i \end{cases}$$

para todo $i = 1, \ldots, r$.

El siguiente paso es calcular un cifrado aleatorio de 1, esto es, $a_r = C_k(1)$, y sucesivamente las diferencias $a_i =$

$a_r - (C^\star_{1,0})_i a_{i+1}$, donde $(C^\star_{1,0})_i$ representa la i-ésima coordenada de la r-tupla $C^\star_{1,0}$, para $i = r-1, \ldots, 1$. Así, no es difícil comprobar que la r-tupla $\delta = (\delta_r, \ldots, \delta_1) = ((C^\star_{1,0})_r, a_{r-1}(C^\star_{1,0})_{r-1}, \ldots, a_1(C^\star_{1,0})_1)$ se corresponde con una r-tupla $(C_k(\epsilon_r), \ldots, C_k(\epsilon_1)$, donde

$$\epsilon_i = \begin{cases} 1, & \text{si } i = \max_{j=1,\ldots,r}\{j \mid \gamma_j = 1\} \\ 0, & \text{en caso contrario} \end{cases}.$$

El sensor S_1 calcula el producto escalar de los dos vectores δ y C_0, llamémosle $\Delta = \delta \cdot C_0$, obtendrá $C_k(1)$ si $m_0 > m_1$ y $C_k(0)$ si $m_0 \leq m_1$. Así pues, si calcula $\Delta C_0 - (a_r - \Delta)C_1$, obtendrá como resultado $C_k(f(m_0, m_1))$, donde $f(m_0, m_1) = \max\{m_0, m_1\}$.

Finalmente, el sensor S_1 envía al resto de sensores de la red $\Delta C_0 + (a_r - \Delta)C_1$, que serán comparados con los propios valores de los sensores de la red.

IV. Conclusión

Encontrar el valor máximo o mínimo de entre un conjunto de valores de una red de sensores puede ser de gran interés, especialmente en aplicaciones donde los sensores toman medidas en un entorno concreto, ya sea, por ejemplo, un sistema de monitorización de incendios o sensores de temperatura o presión. Sin embargo no se conocen demasiadas propuestas que propongan una solución satisfactoria a dicho problema.

En este artículo proponemos una solución. Para ello, nos centramos en comparar dos a dos, los valores de dos sensores. Extendiendo reiteradamente al resto de sensores de la red, obtenemos el resultado deseado. Para ello, hemos utilizado los homomorfismos de privacidad, que ya han sido especialmente utilizados en la literatura en el escenario de las redes de sensores inalámbricas. Especialmente, utilizamos el homomorfismo de privacidad de Domingo-Ferrer, que permite, no sólo calcular la suma de los mensajes en claro a partir de los respectivos mensajes cifrados, si no también la multiplicación. Nuestra propuesta representa una reducción logarítmica en la medida del mensaje respecto a previas propuestas.

Pese a la reducción de la longitud de los mensajes, la propuesta que presentamos aquí continúa siendo compleja para ser eficientemente utilizada en una red de sensores inalámbrica. Así pues, protocolos alternativos más eficientes son una línea de trabajo futuro. Aplicaciones relacionadas, como por ejemplo calcular aquellas regiones donde los sensores miden un valor superior a un valor prefijado que se lanza al inicio del protocolo serán también objeto de estudio por los autores.

Agradecimientos

Nos gustaría agradecer a los anónimos revisores sus comentarios, que han servido para mejorar la presentación de este artículo.

References

[1] C. Aguilar Melchor, P. Gaborit, J. Herranz, "Additively Homomorphic Encryption with t-Operand Multiplications", e-print IACR.

[2] M. Acharya, J. Girao, D. Westhoff, "Secure Comparison of Encrypted Data in Wireless Sensor Networks", En Actas del Proceedings of the Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, IEEE Computer Society, págs.47–53, 2005.

[3] R.Agrawal, J. Kiernan, R. Srikant, Y. Xu, "Order-Preserving Encryption for Numeric Data", SIGMOD Conference 2004, págs. 563–574, 2004.

[4] D. Boneh, E. J. Goh, K. Nissim, "Evaluating 2-DNF formulas on ciphertexts", en Actas del Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, LNCS 3378, págs. 325–341.

[5] J. Domingo-Ferrer, "A provable secure additive and multiplicative privacy homomorphism", en *Information Security Conference*, LNCS 2433, págs. 471–483, 2002.

[6] L. Ertaul Vaidehi, "Computing Aggregation Function Minimum/Maximum using Homomorphic Encryption Schemes in Wireless Sensor Networks (WSNs)", 2008.

[7] J. Girao, D. Westhoff, Mithun Acharya, "Concealed Data Aggregation for Reverse Multicast Traffic in Sensor Networks: Encryption, Key Distribution, and Routing Adaptation", IEEE Transactions on Mobile Computing, **5** (10), págs. 1417–1431, 2006.

[8] J. Girao, D. Westhoff, and M. Schneider, "CDA: Concealed Data Aggregation in Wireless Sensor Networks", En Actas de *IEEE Int'l Conf. Comm. (ICC '05)*, May 2005.

[9] D. Naccache, J. Stern, "A New Public-Key Cryptosystem", en Actas de Eurocrypt'97, LNCS 1233, págs. 27–36, 1997.

[10] T. Okamoto, S. Uchiyama, "A New Public-Key Cryptosystem as Secure as Factoring", en Actas de Eurocrypt'98, LNCS 1403, págs. 208-318, 1998.

[11] R. Rivest, L. Adleman, M. Dertouzos, "On data banks and privacy homomorphisms", en *Foundations of Secure Computation*, págs. 169–180. Academic Press, 1978.

[12] D. Wagner, "Cryptanalysis of an Algebraic Privacy Homomorphism", en Actas de Sixth Information Security Confence, (ISC03), LNCS 2851, págs. 234–239, 2003.

[13] D. Westhoff, J. Girao, M. Acharya, "Concealed Data Aggregation for Reverse Multicast Traffic in Sensor Networks: Encryption, Key Distribution, and Routing Adaptation", *IEEE Transactions on Mobile Computing*, **5** (10), págs, 1417–1431, 2006.

Clasificación de las amenazas a la seguridad en sistemas RFID - EPC Gen2

Joan Melià-Seguí*, Joaquin Garcia-Alfaro*†, Jordi Herrera-Joancomartí‡

* Universitat Oberta de Catalunya, Rambla de Poblenou 156, 08018, Barcelona
† Institut Telecom, Telecom Bretagne, 35576, Cesson-Sevigne, France
‡ Universitat Autònoma de Barcelona, Edifici Q, 08193, Bellaterra

***Abstract*—La tecnología EPC (Electronic Product Code) está basada en la utilización de radio-etiquetas de bajo coste. El uso de estas radio-etiquetas proporciona una gran flexibilidad para la identificación de objetos en movimiento en cadenas de suministro y de producción industrial. Sin embargo, la carencia de mecanismos específicos de seguridad que garanticen propiedades tan indispensables como autenticación o confidencialidad no se recogen actualmente en las especificaciones del estándar EPC. Por ello, es difícil hoy en día hablar del uso de esta tecnología sin que nos venga a la mente problemas de seguridad y de posibles violaciones a la privacidad de sus usuarios. Presentamos en este artículo una vista rápida a la familia de amenazas a las que se enfrenta la tecnología EPC.**

***Index Terms*—RFID, EPC Gen2, modelo de adversario, seguridad, privacidad.**

I. Introducción

La tecnología EPC (del inglés, *Electronic Product Code*), se basa en la utilización de dispositivos RFID (*Radio Frequency IDentification*) [1]. Esta tecnología está destinada a ser la sucesora de los hoy omnipresentes códigos de barras. Diseñada en los laboratorios Auto-ID del MIT (*Massachusetts Institute of Technology*) y más adelante desarrollada por el consorcio EPCglobal Inc., la tecnología EPC representa el elemento clave de una arquitectura distribuida conocida como EPCglobal Network. Los elementos principales de un sistema RFID son las etiquetas electrónicas, los lectores y los sistemas de información (servidores y bases de datos). El objetivo de esta arquitectura es la identificación automática de objetos en movimiento en cadenas de suministro y de producción industrial.

Las etiquetas electrónicas del sistema EPC, cuyas características principales se detallan en la Tabla I, son pasivas (se alimentan del campo eléctrico generado por el lector, debido a la ausencia de batería en la etiqueta). Funcionan en la banda *Ultra High Frequency* (UHF), siendo en Europa entre los 865 y 868 MHz. El rango de lectura entre lector y etiqueta se sitúa alrededor de los 5 metros. Su funcionamiento responde al modelo de una máquina de estados. En un sistema RFID de bajo coste como el EPC Gen2 las etiquetas electrónicas tienen una capacidad muy limitada, permitiendo reducir su coste por debajo de los 10 céntimos [3], pero a su vez, con severas limitaciones para gestionar las amenazas de seguridad.

Como sucede en otras tecnologías emergentes, la falta de seguridad y las amenazas contra los componentes de la arquitectura pueden comportar múltiples inconvenientes a sus usuarios (por ejemplo, difusión de datos privados y pérdida de intimidad). El presente artículo se centra en las amenazas a la integridad de las comunicaciones entre los lectores y las etiquetas electrónicas, debido a la limitación de las etiquetas en el sistema EPC Gen2, y al uso de un canal inalámbrico inseguro que no garantiza la autenticidad de las entidades participantes en el sistema [4].

TABLE I
Principales características de la tecnología EPC Gen2

Identificador	96 bits
Rango de lectura	$\sim$ 5 m
Consumo etiquetas	$\sim$ 10 μW
Frecuencia	865-868 MHz (UHF)
Ratio Tx etiquetas	40 - 640 kbps
Ratio Rx etiquetas	26.7 - 128 kbps
Identificaciones por segundo	$\sim$ 200

El análisis de las vulnerabilidades relativas a la comunicación entre el lector y el sistema de información no se considera en este artículo, puesto que estos equipos tienen la potencia suficiente para ejecutar los mecanismos de cifrado necesarios, además de utilizar canales de comunicación más seguros como interfaces cableadas de red local.

De este modo para el resto del artículo nos referimos a la comunicación entre etiquetas y lectores EPC por el canal de radiofrecuencia como sistema EPC Gen2 , en donde se encuentran la mayoría de las vulnerabilidades de seguridad.

El modelo de comunicación en el sistema EPC Gen2 es común al del resto de sistemas RFID de bajo coste, en donde el lector inicia la comunicación (ITF *interrogator talks first*, del inglés) ya que la etiqueta es pasiva y necesita de la energía del lector para responder. Concretamente existen tres etapas en la comunicación entre un lector y etiqueta EPC Gen2. En las etapas de *selección* e *inventariado*, el lector inicia la comunicación lanzando una petición de identificación (*request*). Las etiquetas presentes en el rango de lectura responden (*response*) con un identificador provisional. En el momento en que el lector responde al identificador provisional (*acknowledge*), las etiquetas devuelven el identificador completo de 96 bits [1]. En este punto si el lector quiere acceder a contenidos de la memoria

reservada, o modificar partes de la memoria de la etiqueta, se entra en la etapa de *acceso*. La Sección II profundiza en esta etapa, en que la comunicación lector-etiqueta se cifra para no revelar datos sensibles del sistema, mientras que el canal etiqueta-lector se transmite sin cifrar.

Como se ha comentado previamente, la característica principal del sistema EPC Gen2 es la simplicidad y bajo coste de las etiquetas electrónicas. Este punto es determinante para la inclusión de posibles mejoras de seguridad que den respuesta a las amenazas que se detallarán en la Sección III, debido a las vulnerabilidades relacionadas con la restricción de la capacidad de computación, memoria y energía, presentes en el sistema EPC Gen2 [5].

En este artículo, presentamos una visión general sobre la familia de amenazas contra la tecnología EPC. Describimos el modelo de adversario del sistema EPC Gen2, las principales vulnerabilidades existentes en la tecnología y los puntos que podrían ser explotados por un atacante para hacer efectivas las amenazas. El resto del artículo se organiza de la siguiente manera. La sección II presenta el modelo de adversario que supondremos durante la presentación de las vulnerabilidades, y las peculiaridades de la tecnología EPC que hacen posible suponer dicho modelo. La sección III describe una presentación general sobre el conjunto de amenazas seleccionado para nuestro estudio. La sección IV cierra el artículo con un conjunto de conclusiones.

II. Mecanismos de seguridad en EPC Gen2

Todo sistema de comunicaciones padece amenazas relacionadas con la seguridad de la información gestionada por el sistema. Por este motivo es importante determinar la naturaleza de dichas amenazas e identificar los posibles adversarios, para poder analizar las medidas de seguridad a adoptar y en qué circunstancias deben ser implementadas.

Las amenazas relativas a la seguridad y privacidad de los datos transmitidos en un sistema EPC Gen2, vienen dadas por el valor intrínseco del objeto etiquetado, o del valor derivado de correlacionar la información de la etiqueta con la identidad del individuo que está identificando [6].

A. Modelo de adversario: definiciones

Para evaluar los principales problemas de seguridad que pueden afectar un sistema EPC Gen2, se debe definir un modelo simple contemplando las características de comunicación del sistema EPC Gen2 de RFID de bajo coste, y los posibles adversarios, así como las capacidades y objetivos de ambos. En primer lugar se listan las principales entidades del sistema EPC Gen2 participantes en el modelo, así como una descripción de sus características. Para un análisis más amplio de modelos de adversario para RFID de bajo coste puede consultarse [7].

- *Lector autorizado*: El que estando registrado en el sistema dispone de los mecanismos necesarios para acceder a los contenidos restringidos de la memoria. Por tanto el lector autorizado puede leer y escribir en las etiquetas electrónicas.
- *Etiqueta legítima*: Una etiqueta electrónica presente en la base de datos del sistema, y que ha sido previamente identificada por un lector autorizado.
- *Lector no autorizado*: El que no está registrado en el sistema, pero tiene acceso al rango de lectura del sistema EPC Gen2.
- *Etiqueta ilegítima*: Etiqueta fraudulenta que accede al rango de lectura de un sistema EPC. Cuando la identificación de una etiqueta fraudulenta ha sido copiada de una etiqueta legítima, se conoce como etiqueta clonada.

A continuación, se definen las características del canal de comunicaciones.

- *Canal lector-etiqueta*: Trasmisión de lector a etiqueta. El lector transmite a una potencia muy superior a la de la etiqueta, ya que esa energía debe ser suficiente para alimentar la respuesta de la etiqueta. Por este motivo, el canal lector-etiqueta puede ser capturado a centenares de metros del punto de transmisión [2].
- *Canal etiqueta-lector*: Transmisión de etiqueta a lector. Como se ha citado en la Introducción, las etiquetas electrónicas del sistema EPC Gen2 son pasivas, por lo que no disponen de una fuente de energía en la propia etiqueta. La transmisión se realiza mediante la señal proveniente del lector reflejado por la antena de la etiqueta, por lo que su alcance se limita a unos 5 metros.

Finalmente, se definen los dos modos de interacción básicos entre lector y etiqueta, la *identificación* y el *acceso*.

- *Identificación*: La etiqueta electrónica legitima o ilegítima transmite (en claro) los 96 bits de su código EPC de identificación tras completar los estados de selección e inventariado.
- *Acceso*: Una vez completada la identificación de la etiqueta electrónica, un lector (autorizado o no autorizado) se dispone a activar los mecanismos de seguridad para poder acceder a todo el contenido de la memoria de la etiqueta para leer o escribir en ella (la Tabla II detalla la esctructura lógica de la memoria). Los peticiones de *acceso* a la memoria de una etiqueta EPC Gen2 pueden ser *read*, *write*, *kill*, *lock*, *access*, *blockwrite*, *blockerase* y *block permalock* [1].

Una vez definidas las características básicas del sistema EPC Gen2, pasamos a describir los posibles tipos de adversarios del sistema. Para el modelo de adversario del sistema EPC Gen2 se supone que las etiquetas y los lectores no autorizados se encuentran, salvo que se indique lo contrario, a una distancia superior a la del rango de lectura del canal *etiqueta-lector*. El motivo por el que este modelo prioriza las amenazas sobre el canal *lector-etiqueta*, es debido a la sencillez de capturar la información de este canal mediante cualquier lector compatible EPC Gen2 a distancias de centenares de metros. El canal *etiqueta-lector*, en cambio, necesita de equipos especiales con antenas muy directivas, o bien situarse dentro del rango de lectura del canal (alrededor de 5 metros).

TABLE II
MAPA LÓGICO DE LA MEMORIA DE LAS ETIQUETAS EPC GEN2

User:	Opcional
TID:	TID [15:0]
	TID [31:16]
EPC:	XPC_W1 [15:0]
	EPC [15:0]
	⋮
	EPC [95:79]
	PC [15:0]
	CRC [15:0]
Reserved:	Access Password [15:0]
	Access Password [31:16]
	Kill Password [15:0]
	Kill Password [31:16]

- *Vulnerabilidad*: Es la propiedad del sistema que un adversario trata de atacar para conseguir el objetivo de la amenaza.
- *Amenaza*: Es el objetivo del adversario para violar una vulnerabilidad relativa a la seguridad del sistema.
- *Adversario pasivo*: Es la entidad que trata de explotar una vulnerabilidad en el sistema para ejecutar la amenaza [8]. Se limita a capturar información en el rango de lectura, sin dar evidencias de su presencia en el sistema.
- *Adversario activo*: Igual que el adversario pasivo, pero puede transmitir y recibir información en el rango de lectura. En caso de poder situarse en el rango del canal *etiqueta-lector*, también podría afectar el contenido de la memoria de las etiquetas.

B. Seguridad del sistema EPC Gen2

El protocolo de comunicación del sistema EPC Gen2 se basa en un sistema de petición-respuesta (*request-response*) entre lector y etiquetas electrónicas en tres etapas diferentes (selección, inventariado y acceso), en el que la etiqueta pasa por diferentes estados. La integridad de los mensajes se comprueba mediante un código de redundancia cíclica (CRC) de 16 bits. La *identificación* de una etiqueta se realiza de manera automática por cualquier lector compatible EPC Gen2, sin ningún tipo de autenticación segura por ambas partes. Es decir, cualquier lector puede identificar etiquetas en el rango del canal *etiqueta-lector*.

Por contra, el protocolo de comunicación para EPC Gen2 sí incluye mecanismos básicos de seguridad para la etapa de *acceso* a las etiquetas electrónicas. El estándar EPC Gen2 incluye en sus especificaciones una contraseña de 32 bits para el acceso a la memoria de la etiqueta electrónica. Además el estándar incluye una contraseña de 32 bits para la opción *kill*, que en caso de ser activada permite desactivar el funcionamiento de la etiqueta de forma permanente, o bien desbloquear determinadas partes de la memoria de la etiqueta electrónica previamente bloqueadas, en función de la codificación del comando como *kill* o como *recommission* [1].

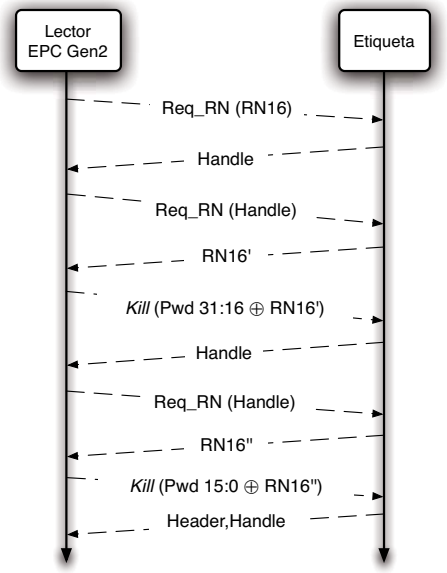


Fig. 1. Protocolo para ejecutar la opción *kill* en EPC Gen2.

Las contraseñas de acceso y *kill* se almacenan en la memoria reservada de la etiqueta electrónica (la Tabla II contiene las diferentes áreas de memoria de una etiqueta EPC Gen2).

Adicionalmente, para evitar revelar información sensible en el canal *lector-etiqueta* (por ejemplo contraseñas o nuevos identificadores) que pudiera ser capturada por un lector no autorizado, las etiquetas electrónicas EPC Gen2 incluyen un generador de números pseudo-aleatorios (PRNG) para cifrar la información transmitida en ese canal. De este modo, cuando el lector requiere el *acceso* a la etiqueta, ésta transmite en plano claves de 16 bits al lector para cifrar el contenido a transmitir mediante una operación de OR-exclusiva a nivel de bit.

Por ejemplo para ejecutar la opción *kill* a una etiqueta EPC Gen2, el lector debe *identificar* previamente la etiqueta. Una vez la etiqueta ha enviado los 96 bits de su código EPC de identificación, el lector procede a la fase de *acceso* (Fig. 1). Para ello el lector solicita a la etiqueta una clave de 16 bits (RN16) como llave de la sesión de *acceso*. Cuando la etiqueta le proporciona el RN16 (representado como *Handle*), el lector solicita una nueva clave (RN16') para iniciar el cifrado de la contraseña de *kill*. Esta operación se repite para las dos partes de la contraseña de *kill* (Pwd [31:16] y Pwd [15:0]) con una nueva clave RN16". Para confirmar la finalización de la operación, la etiqueta transmite un último mensaje formado por una cabecera y el código *Handle*.

III. PRINCIPALES PROBLEMAS DE SEGURIDAD

Tras definir el modelo de adversario para los sistemas EPC Gen2, se pueden destacar las siguientes vulnerabilidades, con la correspondiente amenaza de ser explotadas por parte de un adversario [6]:

- El canal etiqueta-lector es un canal inseguro.
- Cualquier lector compatible con EPC Gen2 puede acceder a la *identificación* y *acceso* de las etiquetas en el rango del canal lector-etiqueta.
- El diseño de las etiquetas está optimizado para reducir su coste, por lo que su capacidad es muy reducida y carece de mecanismos de seguridad y autenticación fiables.

Aunque el contenido de las transmisiones entre lector y etiqueta en el modo de acceso esté cifrado, el hecho de que las claves de cifrado circulen en claro por el canal *lector-etiqueta* representa una vulnerabilidad con riesgo de ser atacada por un adversario.

Por ejemplo el uso de PRNGs con malas propiedades estadísticas, o con cierto grado de predictabilidad, puede suponer riesgos graves en la confidencialidad de las comunicaciones, como se demuestra en [9]. Un lector no autorizado puede acceder el canal *lector-etiqueta* de lectores autorizados y etiquetas legítimas. De este modo un adversario pasivo podría analizar la predictabilidad de las secuencias pseudo-aleatorias generadas en una etapa de acceso. Si el adversario puede obtener información sobre la generación de las secuencias pseudo-aleatorias, le será suficiente con realizar una operación de OR-exclusiva entre la transmisión cifrada y las secuencias predichas, para descifrar el mensaje. De este modo un lector no autorizado en el rango del canal *lector-etiqueta*, obtendría acceso a las zonas de memoria reservadas de la etiqueta tales como contraseñas de acceso y *kill*.

En los siguientes subapartados se detallan las cuatro principales amenazas a la seguridad en un sistema EPC Gen2.

A. Escuchas fraudulentas

Dado que el modelo de comunicación para un sistema RFID pasivo como EPC Gen2 contempla potencias de emisión del lector mucho mayores que la potencia de emisión de las etiquetas, las escuchas fraudulentas se definen como la presencia de lectores no autorizados con acceso a la comunicación del canal *lector-etiqueta*. Esto no excluye la presencia de lectores no autorizados en el canal *etiqueta-lector*, pero es menos probable.

El canal de comunicación entre lectores y etiquetas es fácilmente accesible dada la inseguridad del canal inalámbrico, con lo que la confidencialidad de los datos transmitidos es fácilmente vulnerable. Como se ha visto al inicio de esta Sección, un adversario puede aprovechar la vulnerabilidad de un PRNG predecible para obtener la información almacenada en la memoria reservada de la etiqueta.

Relacionados con las escuchas fraudulentas existen los ataques de *eavesdropping* o análisis, en los que un lector no autorizado podría interceptar la comunicación y analizarla para tratar de descifrar las contraseñas de *acceso* y *kill*.

Por ejemplo un adversario situado en el rango de lectura del canal *lector-etiqueta* de un centro de fabricación textil, en donde se identifica cada producto con una etiqueta EPC, podría recoger información como el número de unidades fabricadas, el modelo o el valor de la producción en cada momento. Si en cambio las etiquetas se utilizan para la identificación de personas, amenazas a la privacidad personal como seguimiento (en inglés, *tracking*) y análisis de perfiles y preferencias (en inglés, *profiling/clustering*) están incluidas en esta categoría [10].

B. Suplantación de identidades

Debido al bajo coste de las etiquetas electrónicas, la tecnología EPC Gen2 se utiliza para la identificación a nivel de objetos o personas [2]. Su diseño está centrado en la simplicidad de sus operaciones, lo que permite *identificar* un gran número de etiquetas de forma simultánea (Tabla I). Puesto que el sistema EPC no dispone de mecanismos de autenticación, el adversario no encontraría ninguna dificultad para conseguir la misma información que podría obtener un usuario autorizado dentro del sistema.

Un lector no autorizado podría suplantar (*spoofing*) un lector autorizado, obteniendo la identificación de etiquetas legítimas. Esta información se podría reproducir en etiquetas ilegítimas o fraudulentas por ejemplo mediante un ataque de *skimming*, lo que significaría un caso de clonación de etiquetas (*cloning*) que podría usarse para falsificación de productos (*counterfeiting*). Un lector autorizado no podría discernir entre una etiqueta legítima y una etiqueta clonada al no existir mecanismos de autenticación para la identificación de etiquetas. Del mismo modo, en un sistema de acceso personal basado en la tecnología EPC Gen2, se podría suplantar la identidad de una persona copiando la identificación de su etiqueta a una etiqueta ilegitima, obteniendo los privilegios de acceso de la persona suplantada.

En el caso que existiera la posibilidad de acceder al rango de lectura del canal *etiqueta-lector*, un lector no autorizado podría realizar ataques activos como *replay* o *scanning* para obtener información de las etiquetas directamente.

C. Divulgación de información

El riesgo derivado de la suplantación de identidad en un sistema EPC Gen2 va más allá de la posible falsificación o clonación de etiquetas electrónicas. Como se ha comentado al inicio de la Sección II, las posibles amenazas a la seguridad y privacidad de un sistema de información están directamente relacionadas con el valor económico de la información que pueda obtenerse.

Esta amenaza es especialmente relevante debido a que el código EPC puede revelar información importante como la marca, el modelo o el precio del producto, así como las estrategias de producción o distribución de la empresa en cuestión. De este modo el adversario puede obtener un beneficio económico de la venta de esta información con fines de espionaje industrial [8].

Divulgar secretos industriales es una actividad claramente ilegal, por lo que el adversario no tomará riesgos innecesarios como acceder a los recintos de producción o distribución. En cambio el adversario puede aprovechar el largo alcance del canal de comunicación *lector-etiqueta* para obtener la información deseada desde centenares de metros [4].

En el plano personal esta amenaza es también relevante, ya que supone una invasión de la privacidad de los usuarios del sistema. Podemos imaginar un escenario en el que la actividad de un grupo de usuarios sea registrada por un adversario con un lector no autorizado, para luego obtener beneficio de esa información.

D. Denegación de servicio

La denegación de servicio (*DoS*) es una amenaza que tiene por objetivo limitar o anular la funcionalidad de un sistema de información. En el caso del sistema EPC Gen2, la denegación de servicio significaría dejar inoperativo el canal de comunicación (tanto de lector a etiqueta como viceversa) haciendo inviable el intercambio de información.

Una denegación de servicio podría realizarse de diversos modos tomando como referencia el modelo especificado en la Sección II. Por ejemplo, un emisor de radio-frecuencia emitiendo señal de ruido (ataque *jamming*) entre las frecuencias 865 y 868 MHz en el rango de lectura del canal *lector-etiqueta*, ocuparía los canales de comunicación del sistema EPC Gen2, impidiendo que los lectores autorizados iniciaran la selección e identificación de las etiquetas electrónicas [1]. Sin ir más lejos, un lector no autorizado compatible EPC Gen2 en el rango de lectura del canal *lector-etiqueta* emitiendo constantemente peticiones de identificación, reduciría considerablemente la eficiencia de lectura de los lectores autorizados, retrasando los procesos de inventariado del sistema atacado.

En el caso de tener acceso al canal *etiqueta-lector*, y haciendo uso de las vulnerabilidades especificadas al inicio de la Sección III se podría realizar un ataque del tipo *tampering*. Un lector no autorizado podría hacer uso del comando *kill* eliminando toda funcionalidad de cualquier etiqueta que entrara en su campo de lectura.

IV. Conclusión

Los sistemas EPC Gen2 representan una de las tecnologías más pervasivas en el ámbito de las tecnologías de la información. La característica principal de la tecnología EPC Gen2 es el reducido precio de las etiquetas electrónicas (previsto por debajo de los 10 céntimos) lo que significa un compromiso entre coste y funcionalidad. Si a ello le añadimos que la comunicación entre etiquetas y lectores se realiza en un canal potencialmente inseguro y que cualquier lector compatible puede acceder a la comunicación de las etiquetas en su rango de lectura, la comunicación del sistema EPC Gen2 padece el riesgo de sufrir ataques a la seguridad y privacidad de sus comunicaciones.

El presente artículo plantea un modelo de adversario en función de las opciones y capacidades del adversario, y de las medidas de seguridad establecidas por el estándar EPC Gen2 (Sec. II). Se hace especial hincapié en la singularidad del modelo de comunicaciones del sistema EPC Gen2 en el que solo se establecen medidas de seguridad para los contenidos transmitidos por el canal *lector-etiqueta*. En la Sección III se hace una descripción detallada de las diferentes amenazas que puede padecer el sistema EPC Gen2, agrupadas en escuchas fraudulentas, suplantación de identidades, divulgación de información y denegación de servicio. Para un análisis más detallado sobre la evaluación de las amenazas relativas a la autenticidad, integridad y disponibilidad de la comunicación en el sistema EPC Gen2, puede consultarse [8].

Agradecimientos

Este trabajo está financiado por el Ministerio de Ciencia y Educación, a través de los proyectos TSI2007-65406-C03-03 E-AEGIS, CONSOLIDER-INGENIO CSD2007-00004 ARES, y una beca doctoral IN3-UOC.

References

[1] EPCglobal. EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860-960 MHz. Tech. report, [On-line] `http://www.epcglobalinc.org/standards/`, 2008.

[2] A. Juels. RFID security and privacy: A research survey. *IEEE Journal on Selected Areas in Communication, vol. 24, no. 2, pp. 381-394*, 2006.

[3] S. Sarma. Toward the 5 cents tag. Auto-ID Lab, Withe Paper, 2001.

[4] G. Avoine and P. Oechslin. RFID traceability: A multilayer problem. In *Financial Cryptography and Data Security, vol. 3570, pp. 124-140, LNCS, Springer Ed.*, 2005.

[5] J. Sounderpandian, R. V. Boppana, S. Chalasani, and A. M. Madni. Models for cost-benefit analysis of RFID implementations in retail stores. *Systems Journal, IEEE, vol. 1, no. 2, pp. 105-114*, 2007.

[6] D. C. Ranasinghe and P. H. Cole. An Evaluation Framework. In *Networked RFID Systems and Lightweight Cryptography, Chapter 8, pp. 157-167, Springer*, 2008.

[7] G. Avoine. Adversarial model for radio frequency identification. Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC, Tech. Rep., 2005.

[8] J. Garcia-Alfaro, M. Barbeau, and E. Kranakis. Handling Security Threats to the RFID System of EPC Networks. In *Security of Self-Organizing Networks: MANET, WSN, WMN, VANET, Auerbach Publications, Taylor &Francis Group*, 2010, in Press.

[9] J. Melia-Segui, J. Garcia-Alfaro, and J. Herrera-Joancomarti. Analysis and improvement of a pseudorandom number generator for EPC Gen2 tags. In *International Workshop on Lightweight Cryptography for Resource-Constrained Devices (Co-located with Financial Cryptography and Data Security 2010 conference)*, LNCS, Springer, 2010.

[10] S. Garfinkel, A. Juels, and R. Pappu. RFID privacy: An overview of problems and proposed solutions. *IEEE Security & Privacy IEEE, vol. 3, no. 3, pp. 34-43*, 2005.

Protocolo de autenticación RFID escalable

Albert Fernàndez-Mir, Jordi Castellà-Roca y Alexandre Viejo
Departament d'Enginyeria Informàtica i Matemàtiques,
UNESCO Chair in Data Privacy,
Universitat Rovira i Virgili, Av. Països Catalans 26,
E-43007 Tarragona, Spain
Email: {albert.fernandez, jordi.castella, alexandre.viejo}@urv.cat

***Resumen*—La identificación por radio frecuencia permite identificar a un objeto de forma remota mediante ondas de radio. Esta característica ha sido utilizada en un gran número de aplicaciones. A partir de esta masiva implantación han surgido problemas de seguridad y privacidad. En este trabajo presentamos un protocolo seguro que protege la privacidad de los usuarios con un bajo coste computacional y que escala correctamente a diferencia de propuestas anteriores.**

I. Introducción

La tecnología RFID (Identificación por Radio Frecuencia) es un sistema que permite la identificación remota de un objeto mediante ondas de radio. Estos sistemas utilizan los siguientes elementos: lectores[1], unos dispositivos llamados etiquetas o en inglés tags[2] que identifican a su portador mediante un número de serie que envían al lector reutilizando la energía de ondas de radio que este emite y un servidor conectado a una base de datos que es el encargado de dar ordenes al lector para comunicarse con las etiquetas.

Actualmente, cualquier artículo es susceptible de incorporar una etiqueta RFID siempre que su precio sea razonable respecto al del artículo y su ciclo de vida permita amortizarlo. Esta generalización esta dando lugar a muchos y novedosos usos de esta tecnología facilitando a sus usuarios un mayor control y una mayor reducción de costes en algunos procesos de producción sobretodo en algunos sectores como el textil o el automovilístico.

Aun así, el hecho de utilizar esta tecnología, conlleva una serie de riesgos para la seguridad y la privacidad de quien la usa. Es por eso que es necesario desarrollar sistemas de seguridad para evitar el uso indebido de la información que estos sistemas manejan tal y como recomienda la Unión Europea en [1]. La parte más vulnerable del sistema, es la comunicación entre el lector y la etiqueta, donde un posible atacante podría obtener, con solo escuchar los mensajes entre estos dos elementos, el identificador de la etiqueta y en consecuencia podría identificar al objeto que la lleva o realizar un seguimiento de la posición o el estado en el que se encuentra de forma indebida.

[1]Los lectores envían ondas de radio mediante una antena incorporada a estos, que permite la comunicación con las etiquetas.

[2]Habitualmente son pasivas, es decir que no disponen de fuente de alimentación propia.

En el siguiente apartado, se analizan los diferentes requisitos que debe cumplir un protocolo de identificación RFID para ser seguro, a continuación en la sección III se describen los diferentes tipos de protocolos de identificación existentes en la actualidad para posteriormente presentar una nueva propuesta en la sección IV que es analizada en la sección V para finalizar con la sección VI donde se muestran las conclusiones.

II. Requisitos de los protocolos RFID

Antes de desarrollar un nuevo protocolo de identificación, es importante tener en cuenta una serie de requisitos que se deben cumplir. En los siguientes apartados se exponen varios aspectos referentes a la privacidad, la seguridad y el rendimiento de un protocolo de identificación RFID.

II-A. Privacidad

Una de las principales preocupaciones en los sistemas RFID es la privacidad. Estos sistemas utilizan las ondas de radio para comunicarse, por lo que cualquier atacante podría escuchar esta comunicación para obtener la identidad de un tag ya que se utiliza un canal de comunicación compartido. Teniendo en cuenta el trabajo de Song et al. en [2] podemos identificar dos cuestiones relativas a la privacidad:

- **Perfil del usuario:** En un sistema RFID tradicional, cuando el lector pregunta al tag, este responde con su identificador único. Si este y otros identificadores correspondientes a un mismo usuario son obtenidos por un lector no autorizado, éste podría realizar un perfil del usuario. Además si los RFIDs contienen información sensible de una persona como por ejemplo un pasaporte o a una tarjeta médica ésta podría revelarse de forma no autorizada.
- **Seguimiento y/o localización:** Si las respuestas de un mismo tag pueden diferenciarse de las de otras etiquetas, un atacante es capaz de obtener su localización mediante diversos lectores situados en puntos estratégicos.

II-B. Seguridad

A continuación se describen los ataques más comunes en sistemas RFID.

- Denegación de servicio: Un atacante puede interceptar los distintos mensajes entre el tag y el lector para impedir que se puedan comunicar.

- Seguimiento del tag: Si un tag se identifica siempre con el mismo identificador, un atacante podría ser capaz mediante varios lectores de saber la localización aproximada de ese tag o de la persona que lo lleve.
- Suplantación de los dispositivos: Un atacante puede hacerse pasar por una etiqueta aún sin conocer su información y comunicarse con los lectores. Por ejemplo en un ataque de replay, un atacante puede escuchar la identificación de otro tag y puede retransmitir el mismo mensaje haciéndose pasar por el tag anterior. De modo parecido, si un atacante conoce el estado interno de una etiqueta, éste es capaz de hacerse pasar por un servidor válido para la etiqueta. Si combinamos estas dos situaciones, nos encontramos con un ataque "Man in the middle" [3].

II-C. Rendimiento

Debido a las limitaciones tecnológicas de los tags RFID hay que tener en cuenta los siguientes requisitos para elaborar un protocolo de identificación.

- **Minimización de la capacidad de almacenaje:** El volumen de datos guardados en el tag debe ser mínimo debido a la limitación de memoria que tienen los tags.
- **Minimización de la capacidad de cálculo:** La computación del lado del tag debe ser mínima ya que no disponemos de mucha energía al ser estos pasivos.
- **Compresión de la capacidad de comunicación:** El volumen de datos que puede transmitir cada tag por segundo esta limitado por el ancho de banda disponible para tags RFID [4], [5].
- **Escalabilidad:** El servidor debe ser capaz de identificar una gran cantidad de tags distintos utilizando el mismo canal de radio. El hecho de utilizar una búsqueda exhaustiva en su base de datos puede ralentizar la identificación.

III. Estado del arte

Existen varios tipos de protocolos de identificación segura para sistemas RFID. En esta sección se describen dos tipos de protocolos basados en criptografia de clave simétrica que a su vez son los más significativos y utilizados en la actualidad. Este problema se podria abordar mediante criptografia de clave asimétrica utilizando por ejemplo curvas elípticas como en [6], pero las capacidades de cálculo de los tags actuales son muy limitadas para implementar este tipo de protocolos.

III-A. Protocolos basados en Hash

Este tipo de protocolos basan su seguridad en las funciones hash unidireccionales, donde a partir del resultado de una función hash resulta computacionalmente inviable obtener su entrada. El primer protocolo basado en hash fue el propuesto por Rivest, Weis, Sarma y Engels en 2003 [7], donde se proponen los hash locks deterministas. Juels en 2006 [8] propuso uno de los protocolos más conocidos en este ámbito, los Hash Locks aleatorios. En este protocolo tanto el lector como el tag generan un número aleatorio que el tag concatena junto a su identificador para generar un hash. Este hash es enviado posteriormente al servidor que busca el identificador que coincida con el resultado de la función hash. El mayor problema de este método es la escalabilidad del sistema [9] ya que es necesario realizar una búsqueda exhaustiva para cada uno de los identificadores hasta encontrar el adecuado. En el mismo sentido trabaja el protocolo OSK [5] que refresca el identificador del tag en cada identificación mediante funciones hash para que éste no pueda ser rastreado. OSK tampoco resulta escalable, por lo que Avoine y Oechslin propusieron una mejora de este protocolo en [11] para mejorar ésta caraterística. Aún así, este tipo de protocolos resultan muy eficientes del lado del cliente (en nuestro caso el tag), ya que las operaciones que debe realizar éste son mínimas lo qual es muy importante debido a la poca capacidad de cálculo de que disponen las etiquetas RFID actuales.

III-B. Protocolos basados en el sincronismo

Estos protocolos surgen debido a la falta de escalabilidad de los protocolos basados en hash. Aunque varios protocolos de este tipo también están basados en hash, tienen la particularidad de utilizar el tiempo de respuesta o un estado de sincronismo entre el lector y el tag. Esto permite en caso de que los dos dispositivos estén sincronizados, realizar una identificación de forma rápida y segura, pero pueden presentar varios problemas cuando un atacante desincroniza los dos dispositivos. Aún así, este tipo de protocolos son escalables en estado de sincronismo, pero presentan problemas cuando se pierde el sincronismo. Uno de estos protocolos, YA-TRAP [10], utiliza una serie de estados de tiempo $[T_0, ..., T_{MAX}]$ que protegen la identidad del tag.

Otro protocolo de este tipo es el propuesto en [12], donde los autores utilizan el estado de sincronismo para realizar una identificación rápida, mientras que utilizan hash locks para identificar los tags cuando estos se encuentran desincronizados. Son comunes los ataques de desincronización en este tipo de protocolos tal y como se indica en [13].

IV. Nuestra propuesta

Este nuevo protocolo consta de tres fases, la fase de inicialización, la fase de identificación y finalmente la fase de actualización que se ejecuta después de cada identificación satisfactoria.

Los objetivos de nuestra propuesta son mejorar la escalabilidad de los protocolos de identificación, y que un atacante no pueda diferenciar si los dispositivos están sincronizados o no. Esto se consigue enviando los mismos bits de información en las dos fases de identificación (principal y secundaria). Además estos bits resultan aleatorios para el atacante en todo momento.

En el cuadro I se muestra la notación utilizada para la descripción del nuevo protocolo de identificación.

id	Identificador del Tag
R	Lector
T	Tag
r_i	Número aleatorio i
$h()$	Función de hash unidireccional
$h_k()$	Función de hash con clave (HMAC)
ks	Clave secreta del lector
$PRNG$	Generador de números pseudo-aleatorios
$SYNC$	Estado de sincronización
C_i	Secuencia de bits pseudo-aleatorios
S	Secuencia de bits pseudo-aleatorios
m_i	Mensaje de actualización i
k_δ	$h_{id}(r_1)$ realizado δ veces
$\|\|$	Operador de concatenación
$\oplus$	Operador de XOR

Cuadro I
NOTACIÓN EMPLEADA EN EL PROTOCOLO

Tabla k		$h'_{id}(r_1)$	r'_1	$h_{id}(r_1)$	r_1	$h_{ks}(id)$	id
k_1							
k_2							
k_3		...	...	...	...	...	...
...							
k_{MAX}							

Cuadro II
TABLA DE VALORES BD

IV-A. Entorno del sistema

Para la implementación de este protocolo es necesario disponer de un servidor con su correspondiente base de datos, y un lector el cual transmita la información obtenida de los diversos tags que haya en el sistema.

Para cada tag, el sistema dispondrá de un registro como el del cuadro II.

Se dispondrá del identificador id y el hash con clave del mismo $h_{ks}(id)$, así como de los valores aleatorios r_1 junto con su hash con clave correspondiente $h_{id}(r_1)$. En este caso guardaremos el r_1 actual y el siguiente r'_1. De esta manera se podrá diferenciar el estado de sincronismo del sistema.

Para añadir un grado más de seguridad en el sistema, el hash del identificador de cada tag es el resultado de un hash con clave (HMAC). ks es la clave que se utiliza en esta operación y que tan solo conocen los lectores autorizados.

IV-B. Fase de inicialización

En esta primera fase, el servidor envía a través del lector la información siguiente: $h_{ks}(id), r_1, h_{id}(r_1)$. Esta información será la utilizada por el tag para realizar la fase de autenticación. El traspaso de esta información se realiza a través de una canal seguro que garantiza la confidencialidad de los datos en el sistema.

IV-C. Fase de identificación principal

En esta segunda fase, ver cuadro III, el lector envía un número aleatorio r_0 al tag, mientras este responde con la información siguiente: $h_{id}(r_1), C_0, r_2$. Donde $C_0 = PRNG(r_1||r_0)$ y r_2 es un número aleatorio generado por el tag. Cuando el lector recibe esta información la envía al servidor para que calcule r_1 y el id. Además, el servidor ha de ser capaz de verificar la secuencia C_0 que el tag ha generado previamente para asegurar que este es auténtico, evitando así una suplantación de identidad o un ataque de replay. Finalmente el tag después de enviar todos los datos pasa a estar desincronizado ($SYNC = 0$) hasta finalizar la fase de actualización.

IV-D. Fase de actualización

En esta tercera fase, el servidor realiza los siguientes cálculos:

$$m_0 = r'_1||h_{id}(r'_1)$$
$$\alpha = h_{ks}(id)||r_2$$
$$m_1 \stackrel{?}{=} h_\alpha(m_0)$$
$$m_2 = m_0||m_1$$
$$S = PRNG(h_{ks}(id)||r_1)$$
$$C_1 = m_2 \oplus S$$

Posteriormente envía C_1 al tag para que este realice las operaciones siguientes:

$$S' = PRNG(h_{ks}(id)||r_1)$$
$$m_3 = C_1 \oplus S'$$
$$\alpha' = h_{ks}(id)||r_2$$
$$m_1 = h_{\alpha'}(m_0)$$
$$r_1 = r'_1, h_{id}(r_1) = h_{id}(r'_1)$$
$$SYNC = 1$$

IV-E. Fase de identificación secundaria

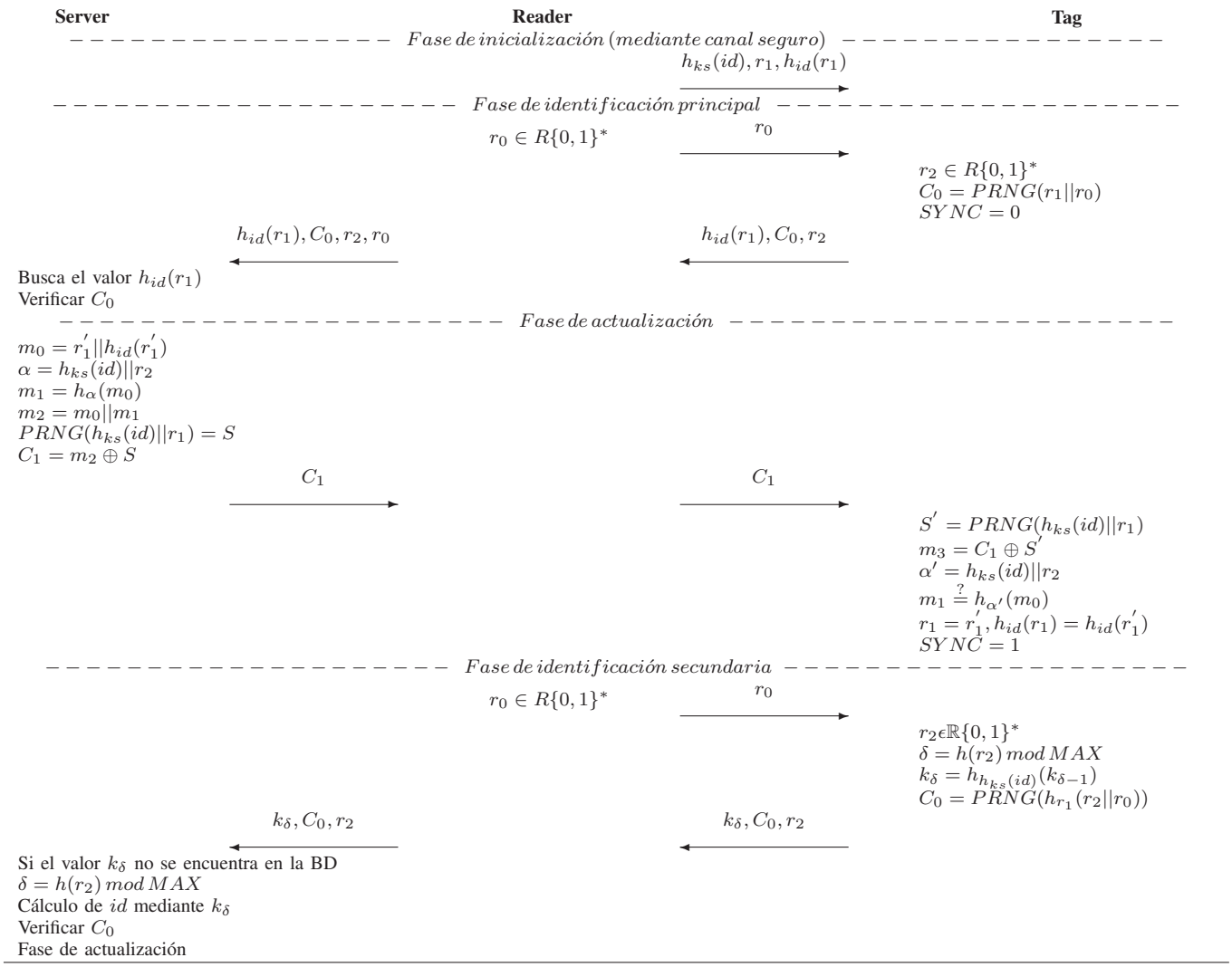
En el caso que C_1 no se reciba correctamente o sea incorrecto, y por lo tanto el estado de sincronización es $SYNC = 0$ el tag seguirá el siguiente protocolo:

1. El lector enviará un r_0 al tag.
2. El tag generará un nuevo r_2, y realizará los siguientes cálculos: para δ dentro de $(1, ..., MAX)$ y siendo $k_0 = h_{id}(r_1)$,

$$\delta = h(r_2) \, mod \, MAX$$
$$k_\delta = h_{h_{ks}(id)}(k_{\delta-1})$$
$$C_0 = PRNG(h_{r_1}(r_2||r_0))$$

y enviará k_δ, C_0, r_2 al lector

1. El lector retransmitirá la información al server que en primer lugar buscará el valor k_δ en la base de datos. Si este valor no se encuentra, se calculará el valor de δ y obtendrá mediante k_δ el identificador correspondiente al tag que será verificado mediante la secuencia C_0.
2. Una vez identificado el tag, el lector volverá a iniciar la fase de actualización.



Cuadro III
PROTOCOLO DE IDENTIFICACIÓN SEGURO

IV-F. Selección del parámetro MAX

El parámetro MAX utilizado influye en el protocolo proporcionado en tres conceptos:

- Espacio de memoria: Considerando n tags, en la BD tenemos una tabla de MAX valores asociados a cada tag. Cada uno de estos valores equivale a la salida de una función hash. Asumiendo la utilización de la función de hash SHA-1 [14], cada uno de estos valores ocupará 160 bits. Como resultado, los requerimientos de memoria para este concepto son: $n \cdot MAX \cdot 160$ bits
- Coste computacional: La fase de actualización secundaria del protocolo requiere localizar k_δ en la tabla de MAX posiciones asociada a cada tag. Una vez localizado, el lector conoce el id del tag correspondiente. El cálculo de δ es previo a dicha búsqueda y por lo tanto la localización de k_δ dentro de la tabla de MAX posiciones se considera un acceso directo (coste $O(1)$). El coste para localizar el tag que contiene k_δ en su tabla es $O(1)$ debido a que sólo debemos saber si éste elemento se encuentra o no en nuestra base de datos. Por lo tanto, el sistema escala correctamente para grandes valores de n (número de tags)
- Seguridad: Tal como se justificará en la sección V referente al análisis de seguridad, cuanto mayor es el valor de MAX mayor es la seguridad proporcionada por el protocolo de identificación propuesto.

Como conclusión, el valor de MAX a elegir dependerá principalmente del espacio de memoria disponible y del tiempo β en que un atacante es capaz de obtener los MAX valores de la tabla k, para el cálculo de este valor se toma una referencia de 200 ms por identificación. Aunque los requisitos de memoria del sistema son elevados, podemos considerar que este recurso es actualmente asequible y por lo tanto dicho requisito es aceptable. En cuanto al tiempo β, éste dependerá del nivel de seguridad que se desee en el sistema. Debido a la potencia actual de los procesadores consideramos

MAX	$Tags$	$Espacio\,en\,disco$	$Tiempo\,\beta$
100	100	0,024 Gb	20 s
1.000	1.000	2,4 Gb	200 s
10.000	10.000	240 Gb	2000 s
100.000	100.000	24 Tb	5,5 h
1.000.000	1.000.000	2400 Tb	55,5 h

Cuadro IV
TABLA DE VALORES MAX

que el tiempo de respuesta al usuario tampoco será un gran inconveniente para el sistema. En la tabla IV podemos observar diversos resultados de la aplicación de valores de MAX y de varios valores de Tags presentes en cada sistema. Fijándonos principalmente en la columna de Tiempo β, podemos observar como para valores elevados de MAX, el tiempo necesario para obtener todos los registros de la tabla k, es de suficiente envergadura para soportar ataques eventuales. Aun así, el precio a pagar por tener un sistema más seguro es el espacio en disco.

V. ANÁLISIS

El protocolo dispone de las siguientes propiedades de privacidad y seguridad:

- **Información privada del tag:** Se garantiza la integridad de la información del tag ya que éste no conoce su propio identificador. Tan sólo conoce $h_{ks}(id)$ por lo que si un atacante consigue averiguar el contenido del tag, no será capaz de diferenciar su verdadero identificador.
- **Localización y seguimiento del tag:** La privacidad de la localización se garantiza ya que en cada nueva identificación, los datos que envía el tag son siempre distintos a los que envió anteriormente ya que $h_{id}(r_1), C_0, r_2$ se actualizan en cada identificación. En el caso que el lector y el tag no estén sincronizados, son k_δ, C_0, r_2 los elementos que varían en cada identificación. La probabilidad que $h_{id}(r_1)$ se repita en estado de sincronismo es de $\frac{1}{2^{160}}$, mientras que la de k_δ es de $\frac{1}{MAX}$ en MAX identificaciones secundarias consecutivas.
- **Suplantación del tag:** No es posible suplantar al tag ya que un atacante no conoce los valores $h_{ks}(id), r_1, h_{id}(r_1)$ con los que se ha inicializado al tag mediante un canal de comunicación seguro y por lo tanto el lector detectaría que no se trata de un tag legítimo y lo rechizaría.
- **Suplantación del servidor:** Análogamente a el apartado de suplantación del tag, no es posible suplantar al servidor a menos que se conozca su base de datos por completo. Suponemos que esta base de datos se encuentra en un entorno seguro.
- **Ataques de replay:** Este tipo de ataques no es posible en este protocolo debido a que en cada identificación cada una de las partes (tag y servidor) proporcionan un nuevo valor obtenido de forma aleatoria. Si un atacante reenvía al servidor un mensaje que haya capturado con anterioridad, éste sólo lo acepta si el mensaje capturado tiene el mismo valor r_0 que le ha enviado previamente. Esto mismo ocurre en el caso del tag, pero con el valor r_2. Los valores aleatorios r_0 y r_2 se generan en cada dispositivo y la probabilidad que se repitan es practicamente nula en un espacio de tiempo corto. En este sentido el valor de $h_{id}(r_1)$ varía en cada identificación principal, mientras que en el caso de la identificación secundaria, es el valor k_δ el que varía hasta MAX veces.
- **Ataques de denegación de servicio:** Este tipo de ataques sí se pueden dar. Aun así podemos entender que un atacante no denegará el servicio para siempre y por lo tanto es un tipo de ataque que puede impedir la identificación de tags pero en ningún caso comporta un riesgo de seguridad en el sistema. Además en el caso que se de este ataque, el sistema no queda bloqueado y sigue funcionando correctamente en identificaciones posteriores. El problema de este tipo de ataque está en el hecho que una vez agotados los elementos de la tabla k el sistema volverá a repetir valores. Aún así si la tabla es suficientemente grande el tiempo en obtener todos los valores puede resultar elevado tal y como se discute en la sección IV-F.

VI. CONCLUSIONES

Después del estudio realizado y de presentar la nueva propuesta, podemos concluir que este nuevo protocolo resulta más escalable que los sistemas basados en hash o las propuestas basadas en sincronismo. Haciendo uso de las dos tecnologías, se consigue obtener un coste O(1) para cualquier identificación. Además, los mensajes enviados en las dos fases de actualización, no permiten a un atacante saber si el sistema esta sincronizado o desincronizado. A este aspecto hay que añadir el hecho que el propio tag no conoce su identificador por lo que si un atacante obtiene los valores con los que ha sido inicializado un tag, no conocerá realmente su identificador en la Base de Datos del sistema.

El aspecto más controvertido del sistema es el espacio utilizado en disco, ya que para un orden de millones de tags, nuestra base de datos se dispara a valores de Terabytes. Aún así este tipo de valores ya son muy comunes en el mercado hoy en día y a un precio razonable, por lo que podremos asumir el coste que supone.

AGRADECIMIENTOS

Los autores agradecen las ayudas del MICINN (proyectos eAEGIS TSI2007-65406-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004), del Ministerio de Industria, Comercio y Turismo (proyecto TSI-020100-2009-720), y del Gobierno de Catalunya (ayuda 2009 SGR 1135). Los autores son responsables de las ideas expresadas en este artículo,

que no reflejan necesariamente la posición de la UNESCO ni comprometen a dicha organización.

REFERENCIAS

[1] Unión Europea, "Recomendación de la comisión sobre la aplicación de los principios relativos a la protección de datos y la intimidad en las aplicaciones basadas en la identificación por radiofrecuencia", en Diario Oficial de la Unión Europea http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:122:0047:0051:ES:PDF, 2009.

[2] Song, B. and Mitchell, C.J., "RFID authentication protocol for low-cost tags", en Proceedings of the first ACM conference on Wireless network securitypp. 140–147, 2008.

[3] Sarma, S. and Weis, S. and Engels, D., "RFID systems and security and privacy implications", en *Cryptographic Hardware and Embedded Systems-CHES*, páginas 1-19, 2002

[4] Avoine, G., "Cryptography in radio frequency identification and fair exchange protocols", en *Faculté Informatique et Communications*, páginas 2007–06, 2005

[5] Ohkubo, M. and Suzuki, K. and Kinoshita, S. and others, "Cryptographic approach to privacy-friendly tags", en *RFID Privacy Workshop*, volumen 82, 2003

[6] Martínez, S. and Valls and M., Roig, C. and Miret, J.M. and Giné, F., "A secure Elliptic Curve-Based RFID Protocol", en *J. Comput. Sci. Tech. 24*, num. 2, páginas 308-318, 2009

[7] R. Rivest S. Weis, S. Sarma and D. Engels, "Security and privacy aspects of low-cost radio frequency identification systems", en *W. Stephan D. Hutter, G. Mller and M. Ullmann, editors, International Conference on Security in Pervasive Computing - SPC 2003, volumen 2802, páginas 454–469. Springer-Verlag, 2003*

[8] A. Juels. "Rfid security and privacy: A research survey.", en *IEEE Journal on Selected Areas in Communications, volumen* 24, número 2, páginas 381-394, 2006

[9] Solanas, A. and Domingo-Ferrer, J. and Martínez-Ballesté, A. and Daza, V., "A distributed architecture for scalable private RFID tag identification", en *Computer Networks*, volumen 51, número 9, páginas 2268-2279, 2007

[10] Tsudik, G. "YA-TRAP: Yet another trivial RFID authentication protocol", en *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, 2006

[11] Avoine, G. and Oechslin P. "A scalable and provably secure hash-based RFID protocol", en *PERCOMW*, páginas 110-114, 2005

[12] Ha, J.C. and Moon, S.J. and Nieto, J. and Boyd, C. "Low-cost and strong-security RFID authentication protocol", en *Emerging Directions in Embedded and Ubiquitous Computing*, páginas 795-807, 2007

[13] Van Deursen, T. and Radomirovic, S., "Attacks on RFID protocols", en *IACR eprint archive 2008*, volumen 310, 2008

[14] Eastlake, D. and Jones, P., "US secure hash algorithm 1 (SHA1)", en *RFC 3174*, setiembre 2001

Criptografía basada en identidad aplicada a los sistemas RFID para mejorar la Seguridad Vial

Jorge Munilla Fajardo
E.T.S. Ingeniería de Telecomunicación
Universidad de Málaga
Email: munilla@ic.uma.es

Andrés Ortiz García
E.T.S. Ingeniería de Telecomunicación
Universidad de Málaga
Email: aortiz@ic.uma.es

Alberto Peinado Domínguez
E.T.S. Ingeniería de Telecomunicación
Universidad de Málaga
Email: apeinado@ic.uma.es

***Abstract*—Desde hace algunos años, la criptografía basada en identidad se ha considerado como una de las propuestas más razonables para los entornos de redes vehiculares (VANETs). Por otra parte, los sistemas que emplean la identificación por radiofrecuencia (RFID) también han sido objetivos de la criptografía basada en identidad. Sin embargo, la combinación de estas dos tecnologías, que permite desarrollar aplicaciones muy interesantes para la seguridad vial, presenta características que dificultan la aplicación de las técnicas criptográficas que se pueden emplear de manera independiente en cada una de ellas. En concreto, las arquitecturas en las que el lector se encuentra localizado en el vehículo y los tags en la carretera presentan unas importantes limitaciones derivadas de la velocidad de los vehículos y de las frecuentes pérdidas de conectividad. En este trabajo se describe este escenario y se presentan los requisitos de seguridad. Asimismo, se propone un esquema general considerando tanto los aspectos criptográficos como los de implementación.**

I. Introducción

Las redes ad hoc vehiculares (VANETs) se pueden considerar actualmente como una nueva generación de redes orientadas a mejorar la seguridad vial y el confort en la conducción. Estas redes permiten la comunicación entre usuarios móviles (vehículos) sin necesidad de acudir a una infraestructura fija que dé soporte a las comunicaciones. El protocolo empleado en estas comunicaciones es el protocolo inalámbrico IEEE 802.11p, en el que los vehículos deben colaborar en las tareas de red para conseguir unos mecanismos eficientes de enrutamiento y transmisión, debido a la topología dinámica de la VANET.

También es habitual la utilización de infraestructuras fijas de apoyo con el fin de mejorar las prestaciones de la red. De esta forma se pueden diferenciar varios tipos de comunicaciones: vehículo a vehículo (V2V) en los casos en los que no se utiliza la infraestructura fija, vehículo a infraestructura (V2I) e infraestructura a vehículo (I2V).

Esta infraestructura no sólo mejora la eficiencia de la red sino que permite el desarrollo de aplicaciones orientadas a la mejora de la Seguridad Vial puesto que los vehículos pueden recibir información sobre el estado del tráfico, y la infrestructura puede obtener información directamente de los vehículos que están circulando [13].

En un entorno VANETs se distinguen dos tipos de información que contribuyen a la mejora de la seguridad vial. En primer lugar, la información originada en los demás vehículos, relacionada habitualmente con la congestión del tráfico o con alertas de accidentes. Esta información se transmite generalmente en modo broadcast [7], [27]. El segundo tipo de información se refiere a la que proviene del entorno, es decir, a la que proporcionan las señales de tráfico, límites de velocidad, peajes o semáforos.

En este sentido, son cada vez más frecuentes las propuestas para dotar al vehículo de sensores que sean capaces de captar de un modo autónomo información sobre su entorno que utilizarán para mejorar su propia conducción y que compartirán con el resto de vehículos a través de enlaces V2V o a través de la infraestructura con enlaces V2I e I2V. Algunas de esas propuestas utilizan técnicas de procesado de imagen para reconocer las señales de tráfico a partir de una imagen tomada por una cámara a bordo del vehículo [18].

Una tecnología que se emplea en las redes vehiculares para que los vehículos se comuniquen con su entorno más cercano es la identificación por radiofrecuencia (RFID). Estos sistemas permiten a un lector detectar la presencia de unas etiquetas (tags) y autenticarlas utilizando frecuencias que pueden estar en banda LF, HF o UHF, principalmente. La ventaja de estos sistemas es que los tags pueden ser pasivos, es decir, no necesitan alimentación. La energía necesaria para transmitir una respuesta al lector la obtienen directamente del campo creado por éste cuando se encuentra a una determinada distancia. Como característica genérica de estos tags hay que destacar su bajo precio, asociado a una limitada capacidad de almacenamiento, y sobre todo, de computación [12].

El modo más sencillo de integrar la tecnología RFID en una VANET consiste en colocar un tag en cada vehículo y uno o varios lectores en lugares estratégicos de la carretera. Este esquema permite implementar comunicaciones V2I con un coste mínimo. Actualmente, casi todas las aplicaciones de la tecnología RFID a las VANETs siguen este esquema, como por ejemplo, el control del peaje [22]. La Unión Europea está trabajando en un sistema RFID de seguimiento para controlar las violaciones de tráfico [1], como aplicación directa de las matrículas electrónicas (Electronic License Plate) [25]. Otras propuestas describen sistemas automáticos para el pago en los aparcamientos y sistemas de prioridad para los semáforos [16].

Sin embargo, la tecnología RFID se puede integrar como parte de los sistemas de adquisición de datos compuestos, principalmente, por sensores a bordo del vehículo. Como

los sistemas RFID no sólo permiten identificar objetos, sino también autenticarlos, suponen una alternativa interesante para establecer comunicaciones del tipo I2V. Por una parte, la RFID no incrementa significativamente el coste de gestión de la VANET, ya que los tags son dispositivos pasivos de bajo coste. Por tanto, se puede desplegar una gran cantidad de tags para complementar a la señalización de tráfico tradicional. La información obtenida a partir de los tags tiene siginificado local debido a la limitación en las distancias de lectura. Además, estos sistemas RFID pueden funcionar en condiciones de climatología adeversa, con visibilidad reducida o nula. Esto permite desarrollar sistemas que informan de la proximidad de intersecciones, de curvas peligrosas, de que se viaja en sentido contrario, etc.

En las siguientes secciones se describe la arquitectura de los sistemas RFID con lectores a bordo de los vehículos y los requisitos de seguridad que deben satisfacer. También se describen los criptosistemas más adecuados para este nuevo escenario que se plantea, concluyendo con unas notas sobre la viabilidad de aplicar criptosistemas basados en identidad.

II. Arquitectura RFID-VANET para mejorar la Seguridad Vial

La integración de los sistemas RFID en VANETs se puede conseguir a través de dos arquitecturas conceptualmente opuestas. Por una parte, la arquitectura *tag a bordo-lector en carretera* que responde al mismo esquema de los sistemas RFID tradicionales que se aplican en otros ámbitos, y por otra la arquitectura *lector a bordo- tag en carretera*. En ambos casos, es necesario utilizar tecnología UHF, debido a las distancias de lectura necesarias.

En la arquitectura *tag a bordo-lector en carretera* los tags tienen movilidad mientras que los lectores permanecen fijos en posiciones determinadas. Esta situación permite que los lectores se puedan conectar con servidores de apoyo que aumenten la capacidad de cálculo o almacenamiento, o que faciliten la funcionalidad requerida en cada momento. En consecuencia, los protocolos de autenticación que se deben aplicar son esencialmente los que se han aplicado hasta ahora.

En cambio, la arquitectura opuesta, denominada *lector a bordo- tag en carretera*, presenta mayores limitaciones o restricciones que dificultan su desarrollo y sobre todo la implementación de protocolos seguros. Una de estas limitaciones queda determinada, fundamentalemnte, por la velocidad a la que se desplazan los vehículos, y en consecuencia, los lectores. Los tags, sin embargo, están colocados en posiciones fijas a lo largo de la carretera proporcionando información con significado local. La otra gran limitación es la pérdida de conectividad que puede sufrir el vehículo durante la marcha por carretera. Esto impide la conexión con servidores externos lo que dificulta enormemente las tareas de autenticación de los tags.

Existen algunas iniciativas que proponen la utilización de esta arquitectura *lector a bordo- tag en carretera* con diversos objetivos [8], [19], [20], [21]. En [20] Pentila et al proponen esta arquitectura para controlar el movimiento de unos robots, es decir, para proporcionar a los robots un sistema de conducción autónoma. Tras realizar pruebas en laboratorio, los autores concluyeron que la máxima velocidad a la que podía funcionar su sistema era de 40 Km/h, lo cual es claramente insuficiente para una VANET. Más tarde, Chon et al [8] proponen esta arquitectura para mejorar el sistema de posicionamiento GPS de los vehículos. En esta ocasión los autores aseguran que la máxima velocidad a la que puede funcionar el sistema es 165 Km/h extrapolando unos cálculos teóricos a partir de unos resultados de laboratorio. Sin embargo, no se han realizado experimentos que lo confirmen. En 2004, una empresa española que participa en el proyecto europeo eSafety [11] lanza una solución comercial para VANETs denominada RBS (road beacon system) [21]. Desgraciadamente no existen datos sobre rangos de velocidad o distancia de trabajo de este sistema. Recientemente, Lee et al [16] han instalado un sistema experimental en un vehículo y han evaluado su funcionamiento. Los resultados establecen que la máxima velocidad de funcionamiento es 100 Km/h con una alta tasa de error en las lecturas.

Aunque estos resultados permiten la utilización del sistema en áreas de velocidad reducida, por ejemplo, en el caso urbano de las ciudades, es necesario seguir investigando, puesto que los datos obtenidos hasta la fecha depende en gran medida de las características particulares de los tags y lectores empleados, que determinan las tasas de transferencias de datos, las distancias máximas de lectura, los tiempos de activación de los tags, etc.

En cualquier caso, y con independencia de las velocidades máximas obtenidas, ninguna de las propuestas existentes tiene en cuenta los requisitos de seguridad, tan necesarios en una implementación orientada a la Seguridad Vial. En la siguiente sección se establecen estos requisitos de seguridad.

III. Requisitos de seguridad de los sistemas RFID-VANET

Los requisitos de seguridad de los sistemas RFID-VANET con arquitecturas *tag a bordo-lector en carretera* son los mismos que los de cualquier sistema RFID tradicional. Esta es la principal razón por la que las aplicaciones RFID que existen actualmente, relacionandas con el tráfico de vehículos, responden únicamente a esta arquitectura. A pesar de ello, muchas de estas aplicaciones no aplican mecanismos de seguridad o presentan un bajo nivel de seguridad, como es el caso de los sistemas de peaje [22].

Aunque se han presentado numerosas propuestas para mejorar la seguridad de los sistemas RFID en banda UHF [6], no se están utilizando porque las aplicaciones reales continuan utilizando el estándar EPC [10], cuya seguridad ha sido comprometida desde hace tiempo. Esto hace que en los casos en los que la seguridad es un requisito real y necesario, se utilicen protocolos propietarios.

Los requisitos de seguridad para los sistemas RFID-VANET con arquitecturas *lector a bordo-tag en carretera* son los siguientes:

Confidencialidad. No es necesario que la comunicación entre el lector y los tags tenga que ser cifrada. Dado que los datos obtenidos de los tags proporcionan información sobre el estado de la carretera, se considera información pública que debería estar disponible para todos los vehículos. Como es lógico, se mantiene el criterio general aplicado al resto de aplicaciones de seguridad vial que se implementan en una VANET [13], [26].

Trazabilidad. Debido a que los tags están colocados en lugares fijos de la carretera, la trazabilidad no supone un problema. Este hecho simplifica notablemente los protocolos de identificación y autenticación.

Autenticación. Como en cualquiera otra arquitectura RFID, la autenticación es la pieza fundamental del sistema. Por tanto, es imprescindible. En la arquitectura RFID-VANET que se analiza en este trabajo, el principal riesgo reside en la posibilidad de que un atacante falsifique algún tag y lo introduzca en el sistema generando falsas informaciones a los vehículos que las detectan, provocando desorientación y desconcierto en los conductores, e incluso accidentes. En consecuencia, se necesita un esquema robusto de autenticación, que resista posibles ataques por repetición y que detecte la clonación de tags.

No-repudio. Este servicio de seguridad no es un requisito para este tipo de aplicaciones. Sin embargo, si el esquema de autenticación empleado es suficientemente seguro, se podría implementar un servicio de no-repudio que permita demostrar a los conductores que estuvieron conduciendo por una determinada carretera, a una cierta hora de un día concreto.

Disponibilidad. Este es siempre un requisito de los sistemas que utilizan la tecnología RFID. Sin embargo, en la arquitectura que estamos analizando no es imprescindible, puesto que la información proporcionada por los tags no es la única que se puede obtener desde el vehículo. Además, la información obtenida a través de los tags debe ser considerada en todo momento como complementaria y el objetivo es ayudar al conductor a tomar una determinada decisión. En ningún caso se pretende que la conducción del vehículo dependa exclusivamente de los datos recopilados por el sistema RFID.

Todos estos requisitos de seguridad simplifican aparentemente los esquemas de identificación a utilizar puesto que únicamente la autenticación es imprescindible. Además, los algoritmos de anticolisión no resultan necesarios ya que los tags serán siempre detectados de uno en uno [12].

Sin embargo, los actuales protocolos de autenticación en RFID no se pueden aplicar directamente debido a una serie de limitaciones: la velocidad a la que circulan los vehículos, las distancias de lectura y la pérdida de conectividad del vehículo con redes externas. Además, aunque la comunicación entre tag y lector tiene una duración muy corta en el tiempo, es importante resaltar que cualquier atacante puede acercarse a uno de los tags ubicados en la carretera e interrogarlo por tiempo indefinido.

Todos estos requisitos y limitaciones determinan no sólo el tipo de tag a utilizar sino también el sistema de seguridad. Así pues, los tags RFID más simples, conocidos habitualmente como tags básicos, y que se emplean para identificación sin autenticación, no son aptos para estas arquitecturas puesto que no incorporan servicio de seguridad alguno. Estos tags permitirían a cualquier atacante no experto falsificar los tags sin más que leer desde un vehículo el identificador que emiten como respuesta.

Los tags que incorporan funciones criptográficas se pueden dividir en dos categorías [15]: tags de clave simétrica y tags de clave asimétrica.

Los primeros no resultan adecuados debido a la enorme complejidad de la gestión de claves. No resulta viable que todos los vehículos conozcan a priori las claves de todos los tags distribuidos por todas las carreteras, ni los tags pueden conocer las claves de todos los vehículos. La razón, además del elevado número de tags, es el hecho de que los vehículos no disponen de una conexión permanente que les permita acceder a un servidor externo.

Los tags de clave asimétrica podrían ser una solución ya que el lector a bordo del vehículo podría verificar mediante la correpondiente clave pública las firmas generadas por los tags. Es decir, si todos los tags tienen asociado un par de claves (pública y privada), el contenido de los tags podría estar protegido mediante la firma realizada con la correspondiente clave privada. El gran inconveniente es la limitada capacidad computacional de los tags, que hace inviable una implementación de estos algoritmos, junto con la reducida capacidad de almacenamiento. Nótese que para verificar la firma de un tag sería necesario leer el certificado de su clave pública, y obtener posteriormente la firma de algún dato que le envíe el lector. De otra manera, si el tag devuelve siempre el mismo mensaje firmado, se podría clonar el contenido incluyendo la firma y crear tags con el mismo contenido.

Dentro de esta categoría de criptosistemas asimétricos hay un tipo especial de sistema conocido como esquema de cifrado o de firma basado en identidad, diseñado para reducir la complejidad global utilizando la propia identidad de los usuarios (p.e. una dirección de correo electrónico) como clave pública, en lugar de los certificados emitidos por una autoridad de certificación [2].

La implemenatación de estos sistemas requiere la existencia de un servidor central de confianza PKG. Este servidor genera su clave privada (maestra) y su clave pública. A continuación el PKG genera la clave privada de cada usuario (bajo petición de cada uno) asociada con la identidad del propio usuario. En el caso de los sistemas RFID, esta clave privada puede ser incorporada a cada tag y a cada lector antes del despliegue del sistema.

La principal ventaja de estos sistemas reside en el modo en que un usuario obtiene la clave pública de otro usuario. En los sistemas asimétricos tradicionales las claves públicas deben ser recuperadas de algún repositorio público. En el caso de los sistemas basados en identidad, un usuario puede generar la clave pública de otro a partir de los datos identificativos de éste último. Por tanto, no es necesario establecer ninguna conexión para verificar las firmas de los tags. Por todo ello, los sistemas basados en identidad parecen los más adecuados para

proporcionar seguridad en las arquitecturas RFID-VANET con lectores a bordo y tags en carretera.

IV. Esquemas basados en identidad para mejorar la Seguridad Vial

La criptografía basada en identidad fue introducida por Shamir en 1984 [23] con un esquema de firma basada en identidad utilizando el algoritmo RSA. En 2001, Boneh y Franklin [3] y Cocks [5] presentaron por vez primera sendos esquemas de cifrado basado en identidad. Desde entonces el interés por estos esquemas ha crecido notablemente.

La criptografía basada en identidad ha sido propuesta en numerosas ocasiones para los protocolos seguros en VANETs. Sin embargo, la primera propuesta para aplicarla en sistemas RFID aparece en 2007. Más concretamente, en [14] se propone la utilización del esquema de firmas cortas de Boneh et al [4] para verificar la autenticidad del contenido de los tags de un sistema RFID de HF que utilizaba etiquetas iCODE de Phillips.

El algoritmo propuesto en [14] consiste en utilizar la identificación del tag como clave pública. El PKG correspondiente genera una clave privada con la que se firma un mensaje que también se almacena en el tag. Por tanto, cuando un lector interroga a un tag, obtiene la identificación del tag, el mensaje que se ha firmado y la firma para que se pueda verificar.

En 2008, Liang y Rong [17], proponen la utilización de criptografía basada en identidad para implementar un protocolo de autenticación mutua entre tag y lector, en el que se firman y se cifran datos. Sin embargo, aunque no especifica la banda de frecuencias de trabajo (LF, HF o UHF), este esquema supone que los tags conocen a priori las identidades de los lectores, lo que hace inviable su aplicación en arquitecturas *lector a bordo- tag en carretera* para VANETs.

Todas las propuestas existentes se han definido sobre arquitecturas RFID tradicionales, que no tienen en cuenta las limitaciones de la velocidad y falta de conectividad de los vehículos, y en algunos casos diseñadas para funcionar fuera de la banda de UHF.

Los datos experimentales de [16] obtenidos en pruebas reales con vehículos, indican que la velocidad máxima de trabajo está limitada a unos 100 Km/h. Estos datos fueron obtenidos utilizando el chip EM4222 [9] que funciona en banda UHF, con una capacidad para transmitir 64 bits a 256 Kbps. Además como este chip implementa procedimientos anticolisión, el tag espera un tiempo aleatorio antes de responder. El tiempo máximo de espera es de unos 62.5 msg. Este hecho es uno de los principales factores en la limitación de la velocidad.

Tomando estos datos como referencia, y dado que las firmas cortas empleadas en [14] son de 160 bits, se puede establecer el siguiente esquema de firma basado en identidad para los sistemas RFID con arquitectura *lector a bordo- tag en carretera*, en el que se aplicará el esquema de firmas cortas de Boneh [4].

Inicialización del sistema. Este proceso se realiza con anterioridad al despliegue de los tags y lectores.

- El PKG genera su par de claves pública K_{Pb} y privada K_{Pr}
- La clave pública K_{Pb} se alamcena en todos los tags y lectores
- Cada tag se personaliza con un identificador $IDtag_i$, siguiendo una codificación EPC o similar [10]
- En cada tag se almcena información $Dtag_{local}$ con significado local, es decir, algún dato relacionado con la localización geográfica en la que será colocado
- Los datos $Dtag_{local}$ son firmados con la clave privada que genera el PKG a partir de la identidad del tag. La firma Sig_i resultante es almacenada también en el tag

Identificación de los tags. Esta fase consiste únicamente en la lectura del contenido de los tags. Es importante recordar que la lectura se realiza con el vehículo en movimiento. Cuando el tag es interrogado por el lector, éstos son los datos que entrega:

- Identificación $IDtag_i$
- Datos $Dtag_{local}$
- Firma Sig_i de los datos

Autenticación de los tags. En esta fase el vehículo puede comprobar la autenticidad de los datos leidos, una vez que ha terminado la comunicación con el tag. La autenticación se realiza mediante la verificación de la firma.

- El lector utiliza la identidad $IDtag_i$ y la clave K_{Pb} como clave pública para verficar la firma Sig_i emitida por el tag sobre los datos $Dtag_{local}$
- Es necesario verificar también que el contenido de los datos $Dtag_{local}$ es coherente con la localización geográfica en la que se encuentra el vehículo. Con esto se consigue detectar la utilización de tags fraudulentos.

Finalmente, con objeto de asegurar que el sistema funcione correctamente, se propone la utilización de otro chip RFID de mayor capacidad y velocidad, el ST XRAG2 [24]. Este chip trabaja también en UHF pero dispone de 304 bits que se pueden utilizar en diversas configuraciones, diviendolo en bancos independientes, uno para el codigo EPC y otro para memoria de usuario, o en un solo banco dedicado al codigo EPC. La velocidad máxima a la que transmite es de 640 Kbps. Con estos datos, que mejoran sustancialmente los del chip empleado en los experimentos de [16], se puede llegar a velocidades mayores, sin tener que renunciar a un nivel de seguridad aceptable.

V. Conclusiones

Si bien es cierto que la utilización de criptografía basada en identidad en entornos de VANETs no es algo nuevo, y que ya existían propuestas para aplicarla también a sistemas RFID, este escenario en el que se integran RFID y VANETs no había sido considerado hasta ahora desde un punto de vista criptográfico.

En este trabajo se analizan los requisitos de seguridad de estos nuevos escenarios y se propone la utilización de un esquema particular basado en identidad que permitiría una implementación razonable a partir de datos experimentales.

En definitiva, con este trabajo se pretende mostrar una nueva situacin en la que la criptografa basada en identidad podra ser la solucin una vez resueltos algunos aspectos de implementacin.

En cualquier caso, es necesario avanzar en la implementación de estos nuevos escenarios y obtener nuevos datos experimentales que permitan una mejor aproximación.

Por otra parte, la utilización de tags comerciales reduce considerablemente las prestaciones del sistema puesto que determinados mecanismos que incorporan desde fábrica no son necesarios. Por ejemplo, el modo en que los vehículos van a leer los tags no requiere un procedimiento anticolisión. En consecuencia, los tiempos de espera aleatorios que se producen al comienzo de las transmisiones se podrían acortar o incluso eliminar, aumentando así las velocidades máximas a las que podría funcionar el sistema o aumentando la cantidad de bits que el tag puede enviar, permitiendo la implementación de esquemas más robustos.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Ciencia e Innovación, a través del proyecto MUOVE (TIN 2008-02236/TSI)

REFERENCES

[1] Asset-road project, http://www.project-asset.com
[2] J. Baek, J. Newmarch, R. Safavi-Naini, W. Susilo, “A survey of identity-based cryptography”, en Proc. of the 10th Annual Conference for Australian Unix User Group, pp. 95-102, 2004
[3] D. Boneh, M. Franklin, “Identity-based encryption from the Weil pairing” en Proc. of CRYPTO 2001, LNCS 2139, pp. 213-229, 2001
[4] D. Boneh, B. Lynn, H. Shacham, “Short signatures from the weil pairing” in Proc. ASIACRYPT, 2001
[5] C. Cocks, “An identity based encryption scheme based on quadratic residues”, Cryptography and coding, Proc. of IMA 2001, LNCS 2260, pp. 360-363, 2001
[6] M. Burmester, B. de Medeiros, J. Munilla, A. Peinado, “Secure EPC Gen2 Compliant radio frequency identification”, 8th International Conference on ADHOC-NOW’2009, 22-25 sept. Murcia, España, Lecture Notes in Computer Science, 5793, pp. 227-240, 2009
[7] C. Chiasserini, E. Fasolo, R. Furiato, R. Gaeta, M. Garetto, M. Gribaudo, M. Sereno, A. Zanella, A.,“Smart Broadcast of Warning Messages in Vehicular ad hoc Application” en Workshop Progetto NEWCOM (NoE), Turin, Italy, 2005
[8] H.D. Chon, S. Jun, H. Jung, S.W.An, “Using RFID for Accurate Positioning” en Porc. of 2004 International Symposium on GNSS/GPS, Sydney, Australia, 2004
[9] EM Microelectronic, “EM4222. Read-only UHF RFID IC”, 2005
[10] EPCGlobal standards, http://www.epcglobalinc.org/standards
[11] eSAFETY project, http://www.esafetysupport.org/
[12] K. Finkenzeller, “RFID Handbook” second edition, Wiley, 2002
[13] H. Hartenstein, K. P. Laberteaux, “VANET. Vehicular applications and inter-networking technologies”, John Wiley and sons, 2010
[14] P. Ith, Y. Oyama, A. Inomata, E. Okamoto, “Implementation of ID-based signature in RFID system” en Proc. of Asia-Pacific Conference on Communications, pp. 233-236, 2007
[15] A. Juels, “RFID Security and Privacy: A research survey”, Selected areas in Communications, IEEE Journal, 2006
[16] E.K. Lee, Y.M. Yoo, C.G. Park, M. Kim, M.Gerla,“Installation and Evaluation of RFID readers on Moving Vehicles” en Proc. of VANET’09, Beijing, China, pp. 99-108, 2009
[17] Y. Lian, C. Rong, “RFID system security using identity-based cryptography” en LNCS 5061, pp. 482-489, 2008
[18] V. Moreno, A. Ledesma, A. Sanchís, “A static images based-system for traffic signs detection” en Proceedings of the 24th IASTED international conference on Artificial intelligence and applications, Innsbruck, Austria, pp 445 - 450, 2006
[19] A. Ortiz, A. Peinado, J. Munilla, “A Scaled Test Benh for Vanets using RFID Signalling” in Computational Intelligence in Security for Information Systems, Advances in intelligent and soft computing, 63, Springer-Verlag, pp 53-59, 2009.
[20] K. Penttil, L. Sydnheimo, M. Kivikoski, “Performance development of a high-speed automatic object identification using passive RFID technology” en Proc. Of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA, pp. 4864-4868, 2004
[21] Road beacon system. http://www.roadbeacon.com/
[22] SANS Institute. “Electronic Toll Collection. An Introduction and Brief Look at Potential Vulnerabilities”, 2004.
[23] A. Shamir, “Identity-based cryptosystems ans signatures schemes” en Proc. of CRYPTO’84, LNCS 196, PP. 47-53, 1984.
[24] STMicroelectronics, “XRAG2. UHF, EPCGlobal Class-1 Generation-2, Contactless Memory Chip 432 bit”, January 2006
[25] Vtt technical research center of Finland, http://www.vtt.fi/research/technology/rfid and wireless sensing.jsp
[26] S. Yousefi, M. Mousavi, M. Fathy, “Vehicular ad hoc networks (VANETs): challenges and perspectives”, en 6th International Conference on ITS Telecommunications, pp. 761-766, 2006
[27] H. Wedde, S. Lehnhoff, B. van Bonn, “Highly Dynamic and Scalable VANET Routing for Avoiding Traffic Congestions” en Proceedings of the fourth ACM international workshop on Vehicular ad hoc Networks, MOBICOM 2007.

Gestionando el riesgo de los activos de las PYMES

Luís Enrique Sánchez, Antonio Santos-Olmo
Departamento de I+D+i
SICAMAN Nuevas Tecnologías
Juan José Rodrigo, 4. Tomelloso, Ciudad Real, Spain
Email: {Lesanchez, Asolmo}@sicaman-nt.com

Eduardo Fernández-Medina, Mario Piattini
ALARCOS Research Group. TSI Department
Universidad de Castilla-La Mancha (UCLM)
Paseo de la Universidad, 4 13071 Ciudad Real, Spain
Email: {Eduardo.FdezMedina, Mario.Piattini}@uclm.es

***Resumen*—La sociedad de la información cada vez depende más de los Sistemas de Gestión y Análisis del Riesgo al que se encuentran sometidos sus principales activos de información, y poder disponer de estos sistemas ha llegado a ser vital para la evolución de las PYMES. Sin embargo, este tipo de compañías requiere que estos sistemas estén adaptados a sus especiales características. En este artículo se presenta el método propuesto para realizar un análisis de riesgos simplificado, que sea válido para las PYMES, y enmarcado dentro de la metodología de gestión de la seguridad en las pequeñas y medianas empresas (MSM2-PYME). Este modelo está siendo aplicado directamente a casos reales, consiguiendo así una constante mejora en su aplicación.**

***Palabras Clave*—PYMES; Análisis de riesgos, Activos; SGSI**

I. Introducción

Estudios realizados [1] han demostrado que para que las empresas puedan utilizar las tecnologías de la información y las comunicaciones con garantías es necesario disponer de guías, métricas y herramientas que les permitan conocer en cada momento su nivel de seguridad y las vulnerabilidades que aún no han sido cubiertas. El problema de conocer los riesgos a los que están sometidos sus principales activos se acentúa especialmente en el caso de las pequeñas y medianas empresas, que cuentan con la limitación adicional de no tener recursos humanos y económicos suficientes para realizar una adecuada gestión de sus activos [2]. Pero con la llegada de Internet, para las empresas es cada vez más critico implantar controles de seguridad que les permitan conocer y controlar los riesgos a los que pueden estar sometidas [3].

Algunos autores [4, 5] sugieren la realización de un análisis de riesgos como parte fundamental de la gestión de la seguridad en las PYMES, ya que los propietarios de estos activos deben tener en cuenta que el valor y la sanción de los datos robados o filtrados en una pequeña organización es el mismo que para una grande, y por tanto debe tener controlado el valor de los activos y los riesgos a los que están sometidos. Otros autores [6] proponen la necesidad de desarrollar un nuevo modelo de análisis de riesgos (AR) pero orientándolo directamente a las PYMES, dado que las características de éstas son diferentes que las de las grandes compañías, considerando que el uso de técnicas de análisis y gestión de riesgos, así como el papel de terceros, es necesario para poder garantizar la seguridad del sistema de información de las PYMES.

Estudios centrados en la evaluación de riesgos [7–9] realizados sobre organizaciones en Europa y los EE.UU, revelan que las PYMES se caracterizan por la falta de la dedicación necesaria a la seguridad de las tecnologías de la información, debido principalmente a la asignación de responsabilidades a personal sin la debida formación. Asimismo, la mayoría de las organizaciones carecen de políticas de seguridad y sistemas de evaluación del riesgo, llegando al caso en que el 73% de los encuestados de PYMES del Reino Unido dijo realizar en su casa la evaluación de riesgos. Menos del 10% de los encuestados afirmó usar una herramienta de análisis de riesgos, y ninguno utilizó una guía de referencia como podía ser la ISO/IEC17799 [10]. Esto plantea dudas sobre la manera exhaustiva o eficaz en que pueden haberse realizado dichos análisis.

Como tal, una de las cuestiones derivadas de las conclusiones es la necesidad de obtener nuevas metodologías y modelos de análisis y gestión del riesgo que se adapten a las características especiales de las PYMES [11], con el objetivo de eliminar (o al menos reducir) los inconvenientes y ayudar a estas compañías a evaluar los riesgos a los que sus activos están expuestos y a establecer los controles de seguridad adecuados.

Por lo tanto, y considerando que las PYMES representan una gran mayoría de empresas tanto a nivel nacional como internacional y son muy importantes para el tejido empresarial de cualquier país, [12] creemos que avanzar en la investigación para mejorar los procesos de análisis y gestión del riesgo para este tipo de empresas puede generar importantes aportaciones. Esto puede contribuir a mejorar no sólo la seguridad de las PYMES, sino también su nivel de competitividad. Por este motivo, a los largo de los últimos años hemos trabajado en elaborar un proceso simplificado que permita analizar y gestionar el riesgo de seguridad en las PYMES [13, 14] , y además hemos construido una herramienta que automatiza completamente este proceso [15] , y lo hemos aplicado en casos reales [16], lo que nos ha permitido validar tanto la metodología como la herramienta.

El artículo continúa en la Sección II, describiendo brevemente las metodologías y modelos existentes para el análisis y la gestión del riesgo de la seguridad y su tendencia actual. En la Sección III se introduce brevemente nuestra propuesta de metodología para el análisis y la gestión del riesgo de la seguridad orientada hacia las PYMES. Finalmente, en la

Sección IV concluimos indicando cuál será el trabajo que desarrollaremos en el futuro.

II. Related Work

Con el propósito de reducir las carencias mostradas en el apartado anterior con respecto a la gestión de la seguridad en la PYMES, ha aparecido un gran número de procesos, marcos de trabajo y métodos para la gestión del riesgo cuya necesidad de uso para proteger de forma eficaz los activos de una compañía está siendo cada vez más reconocida y considerada por las organizaciones, pero que no terminan de tener éxito en el caso de las PYMES.

A pesar de ello, la gestión de la seguridad no puede limitarse al análisis y la gestión del riesgo [17], sino que además de identificar y eliminar riesgos el proceso se ha de realizar de manera eficiente, obteniendo la compañía grandes ahorros de costes como consecuencia directa de una mejor gestión de la seguridad [18]. Gracias al análisis de riesgos se podrán identificar los activos y conocer el nivel de seguridad que se debe aplicar.

Los estándares de gestión de la seguridad más destacados han incorporado procesos para el análisis y la gestión del riesgo, pero estos se han mostrado difíciles de aplicar en el caso de las PYMES, ya que requieren una gran inversión y son difíciles de gestionar [19]. Entre las principales propuestas para el análisis y gestión del riesgo podemos destacar MAGERIT [20], OCTAVE [21]] o CRAMM [22].

Por otro lado, algunos de los principales estándares de gestión de la seguridad han intentando incorporar dentro de sus procesos el análisis y la gestión del riesgo:

- ISO/IEC27005 [23]: Establece las directrices para la gestión del riesgo en la seguridad de la información. Apoya los conceptos generales especificados en la norma ISO/IEC27001 [24] y está diseñada para ayudar a la aplicación satisfactoria de la seguridad de la información basada en un enfoque de gestión de riesgos.
- ISO/IEC21827/SSECMM [25]: El modelo de capacidad y madurez en la ingeniería de seguridad de sistemas describe las características esenciales de los procesos que deben existir en una organización para asegurar una buena seguridad en los sistemas, incluyendo en las fases previas un proceso orientado al riesgo, con 4 subprocesos: SSE-PA02 (Determinar el impacto), SSE-PA03 (Identificar los riesgos de seguridad), SSE-PA04 (Identificar las amenazas), SSE-PA05 (Identificar las vulnerabilidades).
- COBIT: Es una metodología para el adecuado control de los proyectos de tecnología, los flujos de información y los riesgos que implica la falta de controles adecuados. Incluye un proceso orientado a evaluar los riesgos, en el dominio PO9. Este proceso se centra principalmente en los criterios de confidencialidad, integridad y disponibilidad, y de forma secundaria en criterios de efectividad, eficiencia, cumplimiento y confiabilidad. Por último, este proceso involucra a diversos perfiles (RRHH, Sistemas de Información, Tecnología, Instalaciones y Datos) involucrados en el sistema de información.

Por otro lado, existe un pequeño conjunto de herramientas de análisis de riesgos. Actualmente las más utilizadas son PILAR y EAR, basadas en Magerit v2 [20]. Otras herramientas utilizadas son la propuesta por ENISA, que incluye un sistema de comparativas, OCTAVES y Octave Automated Tool, que implementan la metodología de evaluación de riesgos OCTAVE [21], CRAMM y COBRA.

El principal problema de estos procesos y herramientas es su complejidad para aplicarlos en el caso de las PYMES, ya que han sido concebidos para grandes empresas [26]. Se justifica en repetidas ocasiones [27, 28] que la aplicación de este tipo de procesos para las PYMES es difícil y costosa. Además, las organizaciones, incluso las grandes, tienden más a adoptar grupos de procesos relacionados como un conjunto que a tratar los procesos de forma independiente [29].

Por lo tanto, y como conclusión de este apartado, se puede decir que es pertinente y oportuno abordar el problema de desarrollar un nuevo proceso para el análisis y gestión del riesgo de la seguridad para los sistemas de información en las PYMES, así como una herramienta que soporte este proceso, tomando como base la problemática a que este tipo de compañías se enfrenta y que ha llevado a continuos fracasos [30] en los intentos de implantación de un SGSI en este tipo de empresas.

III. Gestión del Riesgo de los Activos en las PYMES

Para solucionar los problemas detectados en el análisis y gestión del riesgo a la hora de aplicarlo en las PYMES, se ha desarrollado un nuevo proceso orientado a gestionar el riesgo en este tipo de compañías denominado ARM-PYME, con dos premisas básicas: i) orientado a las PYMES; y ii) enfocado a reducir los costes de generación y mantenimiento del proceso de análisis y gestión del riesgo.

Este proceso se ha obtenido mediante la aplicación del método de investigación en acción [31] y se ha enmarcado dentro de la metodología (MSM2-PYME) [32] que acomete todos los aspectos relacionados con la gestión de la seguridad.

Dentro de la metodología, el proceso que se encarga del análisis y la gestión del riesgo esta formado por dos actividades:

- Actividad I: Se establece una estructura de relaciones entre los diferentes elementos involucrados en el análisis del riesgo y los controles necesarios para gestionar la seguridad. Estas relaciones se establecen mediante el conocimiento adquirido en las diferentes implantaciones, que es almacenado en una estructura denominada esquema para ser reutilizado con posterioridad, reduciendo los costes de generación de este proceso.
- Actividad II: Mediante la selección del esquema más adecuado y la identificación de un pequeño conjunto de los principales activos se obtiene un detallado mapa de la situación actual (análisis del riesgo) y un plan de recomendaciones de cómo mejorarlo (gestión del riesgo).

Para entender correctamente el proceso es importante conocer el concepto de Esquema. Se trata de una estructura formada

por los principales elementos de un SGSI y las relaciones que se pueden establecer entre ellos, mediante el Know-How adquirido en diferentes implantaciones. Esta estructura puede ser reutilizada por un conjunto de compañías con características comunes (mismo sector y tamaño) a partir del conocimiento adquirido con la implantación de la metodología MSM2PYME y posteriores refinamientos.

Este apartado se divide en dos subapartados, que se corresponden con las dos actividades del proceso.

A. ARM-PYME Actividad 1: Análisis de riesgos como parte de un Esquema.

El principal objetivo de esta actividad es seleccionar los elementos necesarios para poder realizar, en actividades posteriores de la metodología, un análisis de riesgos básico y de bajo coste (que se adapte a los requerimientos de las PYMES) sobre los activos que componen el sistema de información de la compañía.

Esta actividad está basada en las conclusiones obtenidas durante la aplicación del método de investigación-acción [31] a diferentes casos de estudio, las cuales han permitido determinar que los elementos que participan en un análisis de riesgos y sus relaciones tienen un alto grado de coincidencia cuando se aplican en PYMES que tienen características parecidas (mismo sector y mismo tamaño), por lo que se pueden establecer dichas relaciones a priori eliminando el coste de tener que analizarlas una por una mediante una labor de consultoría en cada caso. Aún cuando existan diferencias entre unas y otras, éstas son irrelevantes con respecto a la configuración final del SGSI obtenido para el caso de las PYMES, dado que este tipo de empresas priorizan el coste a obtener un resultado con un alto grado de precisión.

En la Figura 1 se puede ver el esquema básico de entradas, tareas y salidas que componen esta actividad:

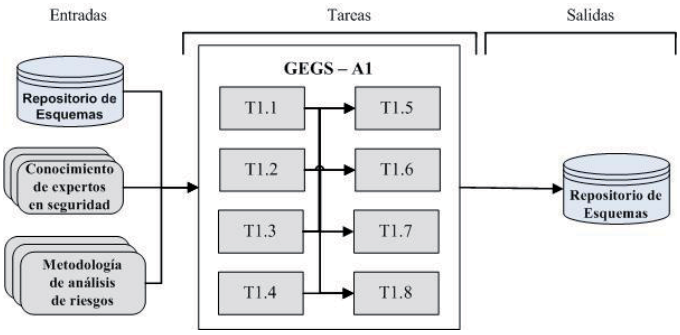


Fig. 1. Esquema simplificado a nivel de tarea de la actividad A1.

- *Entradas:* Como entrada se recibirá el conocimiento del grupo de expertos del dominio de seguridad (EGD) obtenido durante el proceso de implantación de SGSIs, así como un conjunto de controles para la gestión de seguridad que se encuentran almacenados en el repositorio de esquemas y un conjunto de elementos necesarios para elaboración del análisis de riesgos.
- *Tareas:* El subproceso estará formado por ocho tareas que se analizarán en detalle posteriormente.
- *Salidas:* La salida producida por este subproceso consistirá en un subconjunto de los elementos de entrada y las relaciones establecidas entre ellos, los cuáles se almacenarán en el repositorio de esquemas y que se corresponden con la tercera parte de los elementos de los que se compondrá el esquema que se quiere generar.

A continuación, se analizarán las diferentes tareas del proceso que involucran y manipulan los elementos del análisis de riesgos.

- *Tarea T1.1 Selección de tipos de activos:* Se ocupa de seleccionar el conjunto de tipos de activos que formarán parte del esquema que se está construyendo. Los tipos de activos se utilizarán posteriormente para diversas tareas: i) agrupar los activos del sistema de información; ii) se relacionarán con otros elementos del análisis de riesgos para facilitar la automatización del mismo.
- *Tarea T1.2 Selección de amenazas:* Se ocupa de seleccionar el conjunto de amenazas que formarán parte del esquema que se está construyendo. Una amenaza se define como un evento que puede desencadenar un incidente en la organización, produciendo daños materiales o pérdidas inmateriales en sus activos [20]. Estas amenazas se relacionarán en tareas posteriores con otros elementos del análisis de riesgos, con el objetivo de poder automatizar y reducir los costes a la hora de evaluar el riesgo al que están sometidos los activos de un sistema de información.
- *Tarea T1.3 Selección de vulnerabilidades:* Se ocupa de seleccionar el conjunto de vulnerabilidades que formarán parte del esquema que se está construyendo. Una vulnerabilidad se define como una debilidad o falta de control que permitiría o facilitaría que una amenaza actuase contra un activo del sistema que presenta la citada debilidad [20]. Estas vulnerabilidades se relacionarán en tareas posteriores con otros elementos del análisis de riesgos, con el objetivo de poder automatizar y reducir los costes a la hora de evaluar el riesgo al que están sometidos los activos de un sistema de información.
- *Tarea T1.4 Selección de criterios de riesgo:* Se ocupa de seleccionar el conjunto de criterios de riesgo que formarán parte del esquema que se está construyendo. Los criterios de riesgo se definen como aquellos criterios que permiten estimar el grado de exposición a que una amenaza se materialice sobre uno o más activos causando daños o perjuicios a la organización. Estos criterios de riesgo se relacionarán en tareas posteriores con otros elementos del análisis de riesgos, con el objetivo de poder automatizar y reducir los costes a la hora de evaluar el riesgo.
- *Tarea T1.5 Establecer relaciones entre tipos de activos y vulnerabilidades:* Se ocupa de establecer las relaciones existentes entre los elementos que componen el conjunto de tipos de activos y los elementos que componen el con-

junto de vulnerabilidades para un esquema determinado.

- *Tarea T1.6 Establecer relaciones entre amenazas y vulnerabilidades:* Se ocupa de establecer las relaciones existentes entre los elementos que componen el conjunto de amenazas y los elementos que componen el conjunto de vulnerabilidades para un esquema determinado.
- *Tarea T1.7 Establecer relaciones entre amenazas y controles:* Se ocupa de establecer las relaciones existentes entre los elementos que componen el conjunto de amenazas y los elementos que componen el conjunto de controles para un esquema determinado.
- *Tarea T1.8 Establecer relaciones entre tipos de activos, vulnerabilidades y criterios de riesgo:* Se ocupa de establecer las relaciones existentes entre los elementos que componen el conjunto de tipos de activos, los elementos que componen el conjunto de vulnerabilidades y los elementos que componen el conjunto de criterios de riesgo para un esquema determinado.

Las asociaciones de las tareas T1.5-8 se establecen por el grupo de expertos del dominio (EGD) en base al conocimiento adquirido en diferentes implantaciones del SGSI.

B. AGR-PYME Actividad 2: Aplicación del análisis de riesgos.

El principal objetivo de esta actividad es establecer una evaluación de los riesgos a los que se encuentran sometidos los principales activos del sistema de información de la compañía sobre la que se quiere implantar el SGSI, así como proponer un plan al responsable de seguridad (Cu/RS) para gestionar los riesgos de la forma más eficiente posible.

En la Figura 2 se puede ver el esquema básico de entradas, tareas y salidas que componen esta actividad:

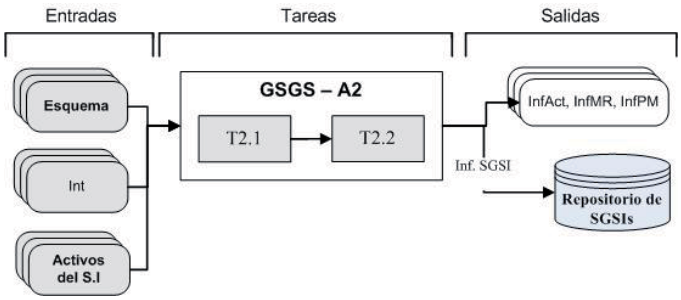


Fig. 2. Esquema simplificado a nivel de tarea de la actividad A2.

- *Entradas:* Como entrada se recibirá: i) un esquema de los existentes en el repositorio de esquemas, que será seleccionado por el consultor de seguridad (SCo) en base a las características de la compañía (sector y tamaño de la misma), del que se obtendrán los elementos necesarios para la realización del análisis de riesgos; ii) el interlocutor (Int) válido para la compañía, que se encargará de definir los activos; iii) un conjunto de activos del sistema de información, lo más generalistas posible (grano grueso).
- *Tareas:* El subproceso estará formado por dos tareas que se analizarán en detalle posteriormente.
- *Salidas:* La salida producida por este subproceso consistirá en una serie de entregables (informe de activos del sistema de información, matriz de riesgos a los que están sometidos los activos del sistema de información y plan de mejora recomendado por la metodología para afrontar las mejoras en la gestión de la seguridad del SGSI) para que el consultor de seguridad (SCo) pueda analizarlos.

El desarrollo de esta actividad está basado en la propuesta de Stephenson [33], que se centra en la sinergia entre la prueba técnica y el análisis de riesgos tomando como referencia la ISO/IEC 27002 [34] y en la metodología de análisis de riesgos Magerit v2 [20]. Estas metodologías suelen producir rechazo en el caso de las PYMES debido a que las perciben como demasiado complejas, a que requieren un enorme compromiso por parte de los miembros de la compañía y a que los costes asociados a los mismos no son aceptados por estas compañías. Por ello, la metodología MSM2PYME simplifica el proceso de evaluación del riesgo para adecuarlo a las PYMES.

Las principales bases sobre las que se define esta actividad son: flexibilidad, simplicidad y eficiencia en costes (humanos y temporales). Se trata, pues, de una actividad que pretende identificar con el menor coste posible los activos de la compañía y los riesgos asociados, usando para ello los resultados generados en las actividades anteriores y unos sencillos algoritmos.

La parte de análisis de riesgos de la metodología desarrollada toma algunos aspectos de Magerit v2 [20] y algunos aspectos de los análisis de riesgos clásicos, pero en todo momento tiende a la simplificación.

Para que esta actividad funcione de forma coherente se deben tener en cuenta las condiciones especiales de las PYMES, en las que los usuarios no suelen tener ni el tiempo ni los conocimientos adecuados para aplicar de forma eficiente metodologías de análisis de riesgos, ni para determinar de forma adecuada los activos de los sistemas de información.

Al igual que en la actividad anterior, cuando se trata de PYMES no se busca la opción óptima sino una opción razonablemente buena que permita grandes reducciones de tiempos a la hora de obtener el resultado.

Las tareas de esta actividad se apoyan principalmente en los datos que componen el esquema seleccionado, generado durante la actividad A1, y en una lista de controles de seguridad.

A continuación mostramos en detalle las tareas que componen la actividad:

- *Tarea T2.1 Identificación de activos:* El objetivo de esta tarea es obtener un conjunto de los activos que componen el sistema de información de la empresa. Los activos definidos son el objetivo principal hacia el que se enfoca el SGSI, ya que son los elementos que se pretenden proteger.

  Una de las diferencias principales que presenta el método para la evaluación del riesgo presentado en la metodología es que se busca que los activos sean lo

más generales posible (grano grueso), frente a [20], que intenta identificarlos de forma clara y precisa (grano fino). En las PYMES se debe intentar definir un conjunto muy pequeño y básico de activos, ya que su sistema de información no permite la protección discriminada de activos de baja atomicidad ni puede soportar el coste de gestión de los mismos. Por lo tanto, en esta tarea se buscarán activos generales que se puedan valorar de forma sencilla tanto desde el punto de vista cuantitativo como cualitativo.

En esta tarea el consultor de seguridad (SCo) deberá ayudar el interlocutor (Int) a identificar el conjunto de activos de valor que componen el S.I. de la compañía.

- *Tarea T2.2 Generación de matriz de riesgos y plan de mejora:* El objetivo de esta tarea es realizar una evaluación de los riesgos a los que están sometidos los activos de la empresa definidos en la tarea T2.1.

  Esta tarea requiere de los datos generados durante la actividad A1 y de los activos identificados en la tarea T2.1 para generar una matriz riesgos que muestre de forma detallada los riesgos a los que está sometido cada activo y un plan de mejora que determine cómo acometer estos riesgos.

  El plan de mejora se soporta sobre los resultados obtenidos de la matriz de riesgos. La matriz de riesgos y el plan de mejora son utilizados por el consultor de seguridad (SCo) para determinar y analizar medidas adicionales y urgentes que deban tomarse en la compañía para mitigar riesgos elevados sobre los activos de información de la misma.

  El primer objetivo de esta tarea es generar una matriz de riesgo que nos permita conocer los riesgos a los que está sometido cada activo de la compañía en cada nivel de madurez y para cada elemento del análisis de riesgos (amenazas, vulnerabilidades y criterios de riesgo). El resultado será una tabla con las siguientes columnas: i) Nivel de Madurez; ii) Nombre y descripción del activo; iii) Coste del activo; iv) Valor estratégico; v) Tipo de activo; vi) Amenaza; vii) Vulnerabilidad; viii) Criterios de riesgo; ix) Nivel de la amenaza (LT); x) Nivel de probabilidad (P); xi) Nivel de riesgo (RL); xii) Nivel de control o cobertura.

El valor obtenido en el nivel de riesgo (RL) se gestionará según la Tabla I y se moverá en un rango comprendido entre 1 (menor riesgo) y 7 (mayor riesgo). Se ha determinado que el nivel del riesgo residual (RRL), es decir, el que tiene actualmente la compañía, nunca debe ser superior al nivel de riesgo aceptable (ARL), que es al que debe tender la compañía. Para el proceso AGR-PYME se ha considerado que el ARL debe ser menor o igual a 3. Si el RL fuera superior al ARL, se procede a la selección de salvaguardas para la reducción del riesgo, realizando el proceso de forma recursiva hasta que el nivel de riesgo de la compañía sea el adecuado.

Para poder obtener de una forma sencilla el riesgo al que está sometido cada activo y el nivel de cobertura de cada control se utilizará el algoritmo de Matriz de Riesgos (RMa).

Tabla I
CUADRO PARA DETERMINAR EL NIVEL DE RIESGO.

ARL=<3	**LT**	**Alto**			**Medio**			**Bajo**		
	P	A	M	B	A	M	B	A	M	B
Valor activo	1	1	2	3	2	3	4	3	4	5
	2	2	3	4	3	4	5	4	5	6
	3	3	4	5	4	5	6	5	6	7

Una vez que se ha obtenido la matriz de riesgos, se utilizará - junto con la información generada en las tareas anteriores - para obtener el plan de mejora, mediante la aplicación del algoritmo del Plan de Mejora (aPM). Este algoritmo funciona de forma recursiva, determinando el activo de mayor riesgo en el menor nivel de madurez, y aplicando el control que permita mejorarlo con el menor coste, para posteriormente recalcular todo el proceso y seleccionar el siguiente mejor, hasta llegar al nivel de gestión de seguridad óptimo.

IV. CONCLUSIONES

En este artículo se ha presentado la propuesta de un proceso para realizar el análisis y gestión del riesgo en las PYMES denominado ARM-PYME, que permite soportar los resultados generados durante la investigación y que cumple con los objetivos perseguidos.

Se ha definido cómo se puede utilizar este proceso y las mejoras que ofrece con respecto a otros modelos que afrontan el problema de una forma más precisa y detallada, pero también más costosa, lo que no los hace válidos para el caso de las PYMES.

Las características ofrecidas por el proceso y su orientación a las PYMES han sido muy bien recibidas, y su aplicación está resultando muy positiva ya que permite a este tipo de empresas realizar una adecuada gestión del riesgo al que están sometidos los activos de su sistema de información. Además, con este proceso se obtienen resultados a corto plazo y se reducen los costes que supone el uso de otros procesos, consiguiendo un mayor grado de satisfacción de la empresa.

El proceso ARM-PYME cumple con los objetivos propuestos, así como con los principios que según la Organización para la Cooperación y el Desarrollo Económico (OECD) [35] debe seguir todo proceso de evaluación del riesgo, según la cuál el sistema debe tener la capacidad de autoevaluar su riesgo de forma continuada en el tiempo, proponiendo medidas.

Finalmente, se considera que el trabajo realizado debe ser ampliado con nuevas especificaciones, nuevos esquemas, mejorando los algoritmos de análisis y gestión del riesgo de forma que puedan ofrecer planes más detallados y profundizando en el proceso con nuevos casos de estudio.

La mayor parte de las futuras mejoras del proceso se están orientando a mejorar la precisión del mismo, pero siempre respetando el principio de coste de recursos, es decir, se busca mejorar el proceso sin incurrir en costes de generación y mantenimiento del análisis de riesgos.

AGRADECIMIENTOS

Esta investigación es parte de los proyectos BUSINESS (PET2008-0136), concedido por el Ministerio de Ciencia e Innovación de España, SEGMENT (HITO-09-138) financiado por la Consejería de Educación y Ciencia de la Junta de Comunidades de Castilla-La Mancha, SISTEMAS (PII2I09-0150-3135) financiado por la Consejería de Educación y Ciencia de la Junta de Comunidades de Castilla-La Mancha MEDUSAS (IDI-20090557) financiado por el Centro para el Desarrollo Tecnológico Industrial. Ministerio de Ciencia e Innovación (CDTI).

REFERENCIAS

[1] Wiander, T. Implementing the ISO/IEC 17799 standard in practice experiences on audit phases. in AISC '08: Proceedings of the sixth Australasian conference on Information security. 2008. Wollongong, Australia.

[2] Wiander, T. and J. Holappa, Theoretical Framework of ISO 17799 Compliant. Information Security Management System Using Novel ASD Method., in Technical Report, V.T.R.C.o. Finland, Editor. 2006.

[3] Dhillon, G. and J. Backhouse, Information System Security Management in the New Millennium. Communications of the ACM, 2000. 43(7): p. 125-128.

[4] Volonino, L. and S. Robinson. Principles and Practice of Information Security. in 1 edition, Anderson, Natalie E. 2004. New Jersey, EEUU.

[5] Michalson, L., Information security and the law: threats and how to manage them. Convergence, 2003. 4(3): p. 34-38.

[6] Spinellis, D. and D. Gritzalis. Information Security Best Practise Dissemination: The ISA-EUNET Approach. in WISE 1:First World Conference on Information Security Education. 1999.

[7] Dimopoulos, V., et al. Approaches to IT Security in Small and Medium Enterprises. in 2nd Australian Information Security Management Conference, Securing the Future. 2004b. Perth, Western Australia: 73-82.

[8] Holappa, J. and T. Wiander, Practical Implementation of ISO 17799. Compliant Information Security Management System Using Novel ASD Method., in Technical Report, V.T.R.C.o. Finland, Editor. 2006.

[9] Llvonen, L. Information Security Management in Finnish SMEs. in 5th European Conference on Information Warfare and Security National Defence College. 2006. Helsinki, Finlan: 1-2 June 2006.

[10] ISO/IEC17799, ISO/IEC 17799, Information Technology - Security Techniques - Code of practice for information security management. 2000.

[11] Taylor, M. and A. Murphy, SMEs and eBusiness. Small Business and Enterprise Development, 2004. 11(3): p. 280-289.

[12] Tawileh, A., J. Hilton, and S. McIntosh, Managing Information Security in Small and Medium Sized Enterprises: A Holistic Approach, in ISSE/SECURE 2007 Securing Electronic Business Processes, Vieweg, Editor. 2007. p. 331-339.

[13] Sánchez, L.E., et al. Security Management in corporative IT systems using maturity models, taking as base ISO/IEC 17799. in International Symposium on Frontiers in Availability, Reliability and Security (FARES06) in conjunction with ARES. 2006. Viena (Austria).

[14] Sánchez, L.E., et al. Developing a model and a tool to manage the information security in Small and Medium Enterprises. in International Conference on Security and Cryptography (SECRYPT07). 2007a. Barcelona. Spain.: Junio.

[15] Sánchez, L.E., et al. SCMM-TOOL: Tool for computer automation of the Information Security Management Systems. in 2nd International conference on Software and Data Technologies (ICSOFT07). . 2007c. Barcelona-España Septiembre.

[16] Sánchez, L.E., et al. Practical Application of a Security Management Maturity Model for SMEs Based on Predefined Schemas. in International Conference on Security and Cryptography (SECRYPT08). 2008. PortoPortugal.

[17] Siegel, C.A., T.R. Sagalow, and P. Serritella, Cyber-Risk Management: Technical and Insurance Controls for Enterprise-Level Security. Security Management Practices, 2002. sept/oct: p. 33-49.

[18] Garigue, R. and M. Stefaniu, Information Security Governance Reporting. Information Systems Security, 2003. sept/oct: p. 36-40.

[19] Bohemer, W. Appraisal of the Effectiveness and Efficiency of an Information Security Management System Based on ISO 27001. in SECURWARE '08: Proceedings of the 2008 Second International Conference on Emerging Security Information, Systems and Technologies. 2008.

[20] MageritV2, Methodology for Information Systems Risk Analysis and Management (MAGERIT version 2). 2006, Ministerio de Administraciones Públicas (Spain).

[21] Alberts, C.J. and A.J. Dorofee, Managing Information Security Risks: The OCTAVE Approach., ed. A.-W.P. Co. 2002.

[22] CRAMMv5.0, CRAMM v5.0, CCTA Risk Analysis and Management Method. 2003.

[23] ISO/IEC27005, ISO/IEC 27005, Information Technology - Security Techniques - Information Security Risk Management Standard (under development). 2008.

[24] ISO/IEC27001, ISO/IEC 27001, Information Technology - Security Techniques Information security management systemys - Requirements. 2005.

[25] ISO/IEC21827, ISO/IEC 21827:2002, Information technology - Systems Security Engineering - Capability Maturity Model (SSE-CMM). 2002, ISO/IEC. p. 123.

[26] Batista, J. and A. Figueiredo, SPI in very small team: a case with CMM. Software Process Improvement and Practice, 2000. 5(4): p. 243-250.

[27] Hareton, L. and Y. Terence, A Process Framework for Small Projects. Software Process Improvement and Practice, 2001. 6: p. 67-83.

[28] Tuffley, A., B. Grove, and M. G, SPICE For Small Organisations. Software Process Improvement and Practice, 2004. 9: p. 23-31.

[29] [29] Mekelburg, D., Sustaining Best Practices: How Real-World Software Organizations Improve Quality Processes. Software Quality Professional, 2005. 7(3): p. 4-13.

[30] Fomin, V.V. and H. Vries. ISO/IEC 27001 Information Systems Security Management Standard: Exploring the reasons for low adoption. in EuroMOT 2008 - The Third European Conference on Management of Technology. 2008. Nice, France.

[31] Kock, N., The threee threats of action research: a discussion of methodological antidotes in the context of an information systems study. , in Decision Support Systems. 2004. p. 265-286.

[32] Sánchez, L.E., et al. MMISS-SME Practical Development: Maturity Model for Information Systems Security Management in SMEs. in 9th International Conference on Enterprise Information Systems (WOSIS07). 2007b. Funchal, Madeira (Portugal). June.

[33] Stephenson, P., Forensic Análisis of Risks in Enterprise Systems. Law, Investigation and Ethics, 2004. sep/oct: p. 20-21.

[34] ISO/IEC27002, ISO/IEC 27002, Information Technology - Security Techniques - The international standard Code of Practice for Information Security Management. 2007.

[35] OECD, OECD Guidelines for the Security of Information Systems and Networks: Towards a Culture of Security., O.f.E.C.-o.a.D. (OECD). Editor. 2002: Paris.

An operational research approach to feature selection for network-based intrusion detection

Hai Nguyen
NISlab, Department of Computer Science
and Media Technology, Gjøvik University College
Email: hai.nguyen@hig.no

Slobodan Petrović
NISlab, Department of Computer Science
and Media Technology, Gjøvik University College
Email: slobodan.petrovic@hig.no

***Abstract*—The goal of the feature selection process in network intrusion detection is to determine the minimal set of network traffic features that ensures accurate intrusion detection in the most efficient way. Obtaining automatically a good set of such features is a great research challenge. The problem is in a huge cardinality of the power set of the full feature set. In this paper, we transform the correlation feature selection (CFS) problem into a polynomial mixed 0-1 fractional programming problem and by solving that problem we get the globally optimal solution of the CFS problem. We describe a sequence of transformations of the original optimization problem into a program with the number of constraints that is linear in the number of full set features. Our feature selection algorithm was compared experimentally with the best-first-CFS and the genetic-algorithm-CFS methods regarding the feature selection capabilities. The classification accuracy obtained after the feature selection by means of the C4.5 and the BayesNet machines over the KDD CUP99 IDS benchmarking data set was also tested. Experiments show that our feature selection method outperforms the best first and the genetic algorithm search strategies by removing much more redundant features and still keeping the classification accuracies or even getting better performances.**

I. Introduction

Network-based intrusion detection systems (IDS) gather and analyze information from networks in order to identify suspicious activities and generate alerts for an operator. Such a task is often analyzed as a pattern classification problem - an IDS has to tell normal from abnormal activities in networks. The theoretical models of IDS (see for example [4], [6]) usually include the representation algorithm (for representing incoming data in the space of selected features) and the classification algorithm (for mapping the feature vector representation of the incoming data to elements of a certain set of values, e.g. normal or abnormal etc.) Some IDS models, like those presented in [6], also include the feature selection algorithm, which determines the features to be used by the representation algorithm. Even if the feature selection algorithm is not included in the model directly, it is always assumed that such an algorithm is run before the very intrusion detection process.

The goal of the feature selection algorithm is to determine the most relevant features of incoming traffic, whose monitoring ensures reliable detection of abnormal behaviour. Since the effectiveness of the classification algorithm heavily depends on the number of features, it is of interest to minimize the cardinality of the set of selected features without dropping potential indicators of abnormal behaviour. Obviously, determining a good set of features is not an easy task. The most of the work in practice is still done manually and the feature selection process depends too much on expert knowledge. Automatic feature selection for intrusion detection remains therefore a great research challenge.

In this paper, we approach the problem of feature selection for intrusion detection from the operational research point of view. We propose an automatic feature selection procedure based on so-called filter method [7], [10] used in machine learning. The filter method directly considers statistical characteristics of the data set, such as correlation between a feature and a class or inter-correlation between features, without involving any learning algorithm. We focus on one of the most important filter methods, the Correlation Feature Selection (CFS) measure proposed by M. Hall [8]. The CFS measure is combined with some search strategies, such as brute force, best first search or genetic algorithm, in order to find the most relevant subset of features. The brute force method can only be applied when the number of features is small. In other cases, a more intelligent optimization algorithm in the feature selection process is needed. With the best first search or the genetic algorithm, we can deal with high dimensional data sets, but these methods usually give locally optimal solutions. To get the globally optimal feature set, we formulate the problem of feature selection by representing the CFS measure as a polynomial mixed 0 - 1 fractional programming (P01FP) problem. We improve the Changs method [1], [2] in order to equivalently reduce this P01FP to a mixed 0 - 1 linear programming (M01LP) problem [1]. Finally, we propose to use the branch-and-bound algorithm to solve this M01LP, whose optimal solution is also the globally optimal subset of relevant features by means of the CFS measure.

Any feature selection algorithm selects relevant traffic features based on labelled data. We used the KDD CUP99 [9] data set for this purpose. The full feature set assigned to this data set consists of 41 features. For evaluating the performance of our feature selection approach, two available feature selection methods based on the CFS measure were implemented [3]. The first one was the best-first-CFS method using the best-first search strategy to find the locally optimal subset of features by means of the CFS measure. The second one

used the genetic algorithm and therefore we call that feature selection approach a genetic-algorithm-CFS method. To test the overall effectiveness of an IDS employing our feature selection algorithm, 10% of the overall (5 million records) KDD CUP99 IDS benchmarking labelled data set were used to train and to test C4.5 [11] and BayesNet [5] machine learning algorithms with 5-fold cross-validation. Experiments show that an IDS applying our feature selection algorithm outperforms the IDS implementing the best first and genetic algorithm search strategies in the feature selection process. Our method removes much more redundant features and the classification accuracies with the reduced feature set are kept at the same level or they become even better.

The paper is organized as follows. Section II describes the CFS measure in more detail. We show how to represent the problem of feature selection by means of the CFS measure as a polynomial 0-1 fractional programming (P01FP) problem. The background regarding P01FP, M01LP problems and Chang's method is also introduced in this section. Section III describes our new approach. We present some experimental results in Section IV. The last section summarizes our findings.

II. Background

A. Correlation feature selection measure

The Correlation Feature Selection (CFS) measure evaluates subsets of features on the basis of two concepts: the feature-classification (r_{cf_i}) correlation and the feature-feature ($r_{f_if_j}$) correlation. The following equation used in [8] gives the merit of a feature subset S consisting of k features:

$$Merit_S(k) = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}. \quad (1)$$

Here, $\overline{r_{cf}}$ is the average feature-classification correlation, and $\overline{r_{ff}}$ is the average feature-feature correlation, as given below:

$$\overline{r_{cf}} = \frac{r_{cf_1} + r_{cf_2} + \ldots + r_{cf_k}}{k}$$

$$\overline{r_{ff}} = \frac{r_{f_1f_2} + r_{f_1f_3} + \ldots + r_{f_kf_1}}{\frac{k(k-1)}{2}}$$

Therefore, we can rewrite (1) as follows:

$$Merit_S(k) = \frac{r_{cf_1} + r_{cf_2} + \ldots + r_{cf_k}}{\sqrt{k + 2(r_{f_1f_2} + r_{f_1f_3} + \ldots + r_{f_kf_1})}}. \quad (2)$$

Suppose there are n full set features. We need to find the subset S of k features, which has the maximum value of $Merit_S(k)$ over all 2^n possible feature subsets:

$$\max_S \{Merit_S(k), 1 \le k \le n\}. \quad (3)$$

We now propose a new method to find the globally optimal subset of features. To this end, we formulate the CFS feature selection task as so-called fractional programming problem. We use binary values of the variable x_i in order to indicate the appearance ($x_i = 1$) or the absence ($x_i = 0$) of the feature f_i in the optimal subset of features. Therefore, the problem of selecting features by means of the CFS measure can be described as a fractional programming problem as follows:

$$\max_{x=(x_1,\ldots,x_n)} F(x) = \frac{\sum_{i=1}^n r_{cf_i}x_i}{\sqrt{\sum_{i=1}^n x_i + \sum_{i\neq j} 2r_{f_if_j}x_ix_j}} \quad (4)$$

or in parameter form:

$$\max_{x=(x_1,\ldots,x_n)} F(x) = \frac{(\sum_{i=1}^n a_ix_i)^2}{\sum_{i=1}^n x_i + \sum_{i\neq j} 2b_{ij}x_ix_j}. \quad (5)$$

In the following subsection, we consider the problem stated above as a polynomial mixed $0-1$ fractional programming ($P01FP$) problem and show how to solve it.

B. Polynomial Mixed $0-1$ Fractional Programming

A general polynomial mixed $0-1$ fractional programming ($P01FP$) problem [2] is represented as follows:

$$\min \sum_{i=1}^m \Big(\frac{a_i + \sum_{j=1}^n a_{ij}\prod_{k\in J} x_k}{b_i + \sum_{j=1}^n b_{ij}\prod_{k\in J} x_k}\Big) \quad (6)$$

subject to the following constraints:

$$\begin{cases} b_i + \sum_{j=1}^n b_{ij}\prod_{k\in J} x_k > 0, i = 1,\ldots,m, \\ c_p + \sum_{j=1}^n c_{pj}\prod_{k\in J} x_k \le 0, p = 1,\ldots,m, \\ x_k \in \{0,1\}, k \in J, \\ a_i, b_i, c_p, a_{ij}, b_{ij}, c_{pj} \in \Re. \end{cases}$$

By replacing the denominators in (6) by positive variables $y_i (i = 1,\ldots,m)$, the $P01FP$ is transformed to the following equivalent polynomial mixed $0-1$ programming problem:

$$\min \sum_{i=1}^m \Big(a_iy_i + \sum_{j=1}^n a_{ij}\prod_{k\in J} x_ky_i\Big) \quad (7)$$

subject to the following constraints:

$$\begin{cases} b_iy_i + \sum_{j=1}^n b_{ij}\prod_{k\in J} x_ky_i = 1; y_i > 0, i = 1,...,m, \\ c_p + \sum_{j=1}^n c_{pj}\prod_{k\in J} x_k \le 0, p = 1,...,m, \\ x_k \in \{0,1\}, k \in J, \\ a_i, b_i, c_p, a_{ij}, b_{ij}, c_{pj} \in \Re. \end{cases} \quad (8)$$

In order to solve this problem, Chang [2] proposed a linearization technique to transfer the terms $\prod_{k\in J} x_ky_i$ into a set of mixed $0-1$ linear inequalities. Based on this technique, the $P01FP$ becomes then a mixed $0-1$ linear programming problem ($M01LP$), which can be solved by means of the branch-and-bound method to obtain the global solution.

Proposition 1: A polynomial mixed $0-1$ term $\prod_{k\in J} x_k y_i$ from (7) can be represented by the following program [1]:

$\min z_i$

subject to the following constraints:

$$\begin{cases} z_i \geq M(\sum_{k\in J} x_k - |J|) + y_i, \\ z_i \geq 0, \end{cases} \tag{9}$$

where M is a large positive value.

Proposition 2: A polynomial mixed $0-1$ term $\prod_{k\in J} x_k y_i$ from (8) can be represented by a continuous variable v_i, subject to the following linear inequalities [1]:

$$\begin{cases} v_i \geq M(\sum_{k\in J} x_k - |J|) + y_i, \\ v_i \leq M(|J| - \sum_{k\in J} x_k) + y_i, \\ 0 \leq v_i \leq M x_i, \end{cases} \tag{10}$$

where M is a large positive value.

We now formulate the optimization problem of the CFS measure (5) as a polynomial mixed $0-1$ fractional programming $(P01FP)$ problem.

Proposition 3: The optimization problem of the CFS measure (5) can be considered as a polynomial mixed $0-1$ fractional programming $(P01FP)$ problem.

Proof. We change the sign of $F(x)$ in (5) to make a minimum problem and decompose the numerator of (5) as follows:

$$(\sum_{i=1}^{n} a_i x_i)^2 = \sum_{i=1}^{n} a_i^2 x_i^2 + \sum_{i\neq j} 2a_i a_j x_i x_j. \tag{11}$$

Therefore, (5) can be written as (6). ■

Remark: By applying the Chang's method, we can transform this $P01FP$ problem to the $M01LP$ problem. The number of variables and constraints will depend on the square of n, where n is the number of features. The reason for this is that the number of terms $\prod_{k\in J} x_k y$, which are replaced by the new variables in forms $\left(\sum_{i\neq j} 2a_i a_j x_i x_j y\right)$ or $\left(\sum_{i\neq j} 2b_{ij} x_i x_j y\right)$, is $n(n-1)/2$. The branch-and-bound algorithm can then be used to solve this $M01LP$ problem. But the efficiency of the method depends strongly on the number of variables and constraints. The larger the number of variables and constraints an $M01LP$ has, the more complicated the branch-and-bound algorithm is.

In the following section, we present an improvement of the Chang's method to get an $M01LP$ with a linear number of variables and constraints in the number of full set variables. We also describe a new search strategy to obtain the relevant subsets of features by means of the CFS measure.

III. Optimization of the CFS measure

By introducing an additional positive variable, denoted by y, we now consider the following problem equivalent to (5):

$$\min\{-F(x)\} = -\sum_{j=1}^{n}(\sum_{i=1}^{n} a_i a_j x_i) x_j y \tag{12}$$

subject to the following constraints:

$$\begin{cases} y > 0, \\ x = (x_1, x_2, ..., x_n) \in \{0,1\}^n, \\ \sum_{i=1}^{n} x_i y + \sum_{j=1}^{n}(\sum_{i=1, i\neq j}^{n} b_{ji} x_i) x_j y = 1. \end{cases} \tag{13}$$

Here, all the terms $a_i a_j x_i x_j$ in the numerator and the terms $b_{ij} x_i x_j$ in the denominator of (5) have been grouped into the sum $\left(\sum_{j=1}^{n}(\sum_{i=1}^{n} a_i a_j x_i) x_j y\right)$ and the sum $\left(\sum_{j=1}^{n}(\sum_{i=1, i\neq j}^{n} b_{ji} x_i) x_j y\right)$, respectively. Each sum contains n terms, which will be equivalently replaced by new variables with constraints following the two propositions given below:

Proposition 4: A polynomial mixed $0-1$ term $(\sum_{i=1}^{n} a_i a_j x_i) x_j y$ from (12) can be represented by the following program:

$\min z_j$

subject to the following constraints:

$$\begin{cases} z_j \geq M(x_j - 1) + (\sum_{i=1}^{n} a_i a_j x_i) y, \\ z_j \geq 0, \end{cases} \tag{14}$$

where M is a large positive value.

Proof.

(a) If $x_j = 0$, then $z_j \geq M(0-1) + (\sum_{i=1}^{n} a_i a_j x_i) y \leq 0$ will force $\min z_j$ to be zero, because $z_j \geq 0$ and M is a large positive value.

(b) If $x_j = 1$, then $z_j \geq M(1-1) + (\sum_{i=1}^{n} a_i a_j x_i) y \geq 0$ will force $\min z_j$ to be $(\sum_{i=1}^{n} a_i a_j x_i) y$, because $z_j \geq 0$.

Therefore, the program on z_j presented above reduces to:

$$\min z_j = \begin{cases} 0, \quad if \quad x_j = 0, \\ (\sum_{i=1}^{n} a_i a_j x_i) y, \quad if \quad x_j = 1, \end{cases}$$

which is the same as $(\sum_{i=1}^{n} a_i a_j x_i) x_j y = \min z_j$. ■

Proposition 5: A polynomial mixed $0-1$ term $(\sum_{i=1, i\neq j}^{n} b_{ji} x_i) x_j y$ from (13) can be represented by a continuous variable v_j, subject to the following linear inequality constraints:

$$\begin{cases} v_j \geq M(x_j - 1) + (\sum_{i=1, i\neq j}^{n} b_{ji} x_i) y, \\ v_j \leq M(1 - x_j) + (\sum_{i=1, i\neq j}^{n} b_{ji} x_i) y, \\ 0 \leq v_j \leq M x_j, \end{cases} \tag{15}$$

where M is a large positive value.

Proof.

(a) If $x_j = 0$, then (15) becomes

$$\begin{cases} v_j \geq M(0-1) + (\sum_{i=1,i\neq j}^{n} b_{ji}x_i)y, \\ \\ v_j \leq M(1-0) + (\sum_{i=1,i\neq j}^{n} b_{ji}x_i)y, \\ \\ 0 \leq v_j \leq 0. \end{cases}$$

v_j is forced to be zero, because M is a large positive value.

(b) If $x_j = 1$, then (15) becomes

$$\begin{cases} v_j \geq M(1-1) + (\sum_{i=1,i\neq j}^{n} b_{ji}x_i)y, \\ \\ v_j \leq M(1-1) + (\sum_{i=1,i\neq j}^{n} b_{ji}x_i)y, \\ \\ 0 \leq v_j \leq M. \end{cases}$$

v_j is forced to be $(\sum_{i=1,i\neq j}^{n} b_{ji}x_i)y$, because M is a large positive value.

Therefore, the constraints on v_j reduce to:

$$v_j = \begin{cases} 0, \quad if \quad x_j = 0, \\ \\ (\sum_{i=1,i\neq j}^{n} b_{ji}x_i)y, \quad if \quad x_j = 1, \end{cases}$$

which is the same as $(\sum_{i=1,i\neq j}^{n} b_{ji}x_i)x_j y = v_j$. ■

We substitute each term $x_i y$ in (13) by new variables t_i satisfying constraints from Proposition 2. Then the total number of variables for the $M01LP$ will be $4n+1$, as they are x_i, y, t_i, z_j and $v_j (i, j = 1, \ldots, n)$. Therefore, the number of constraints on these variables will also be a linear function of n. As we mentioned above, with Chang's method [2] the number of variables and constraints depends on the square of n. Thus our new method improves Chang's method by reducing the complexity of the branch and bound algorithm.

We now present a new search strategy for obtaining subsets of relevant features by means of the CFS measure.

- **Step 1:** Calculate all feature-feature ($r_{f_i f_j}$) and feature-classification (r_{cf_j}) correlations from the training data set.
- **Step 2:** Construct the optimization problem (4) from the correlations calculated above. In this step, we can use expert knowledge by assigning the value 1 to the variable x_i if the feature f_i is relevant and the value 0 otherwise.
- **Step 3:** Transform the optimization problem of CFS to a mixed $0-1$ linear programming $(M01LP)$ problem, which is to be solved by the branch-and-bound algorithm. A non-zero integer value of x_i from the optimal solution indicates the relevance of the feature f_i regarding the CFS measure.

IV. Experimental work

A. Experimental setting

For evaluating the performance of our new CFS-based approach, two available feature selection methods based on the CFS measure [3] were implemented. The first one is the best-first-CFS method, which uses the best first search strategy to find the locally optimal subset. The second one uses the genetic algorithm for search. Note that the best first search and the genetic algorithm do not guarantee to find the globally optimal solution. We can overcome this problem with our new method. We did not choose the exhaustive search method since it is not feasible for feature selection from data sets with a large number of features. We applied machine learning algorithms for evaluating the classification accuracy on selected features, since there is no standard IDS.

TABLE I
THE PARTITION OF KDD CUP'99 DATA SET USED IN THE EXPERIMENT

Classes	Number-of-instances	Percentage
KDD99-normal	97.278	18.30%
KDD99-DoS	391.458	73.74%
KDD99-Probe	41.113	7.74%
KDD99-U2R	52	0.01%
KDD99-R2L	1.126	0.21%
Total	531.027	100%

We performed our experiment using 10% of the overall (5 million records) KDD CUP'99 IDS benchmarking labelled data set [9]. This data set contains normal traffic and four main attack classes: (i) Denial of Service (DoS) attacks, (ii) Probe attacks, (iii) User to Root (U2R) attacks and (iv) Remote to Local (R2L) attacks. The numbers of instances for the four attack classes and the normal class are quite different. For example, the ratio of the number of U2R attacks and the number of DoS attacks is $1.3 * 10^{-4}$. Details on the numbers of class instances are given in Table I.

We tested the performance of our newly proposed CFS-based feature selection method as follows:

1) Feature selection is performed on the basis of the whole data set: (1a) Each attack class and the normal class are processed individually, so that a five-class problem can be formulated for feature extraction and classification with a single classifier. (1b) All attack classes are fused so that a two-class problem can be formulated, meaning the feature selection and classification for normal and abnormal traffic is performed. It might be well possible that the attack-recognition results are not satisfactory for all of the classes, since the numbers of class instances are unevenly distributed. In particular, the classes U2R and R2L are under-represented. The feature selection algorithm and the classifier, which is used for evaluation of the detection accuracy on selected features, might concentrate only on the most frequent class data and neglect the others. As a consequence, we might miss relevant characteristics of the less represented classes.
2) As the attack classes are distributed so differently, we prefered to process these attack classes separately. With the specific application of IDS we can also formulate four different two-class problems. Four classifiers shall be derived using specific features for each classifier in order to detect (identify) a particular attack. The rationale for this approach is that we predict the most accurate classification if each of the four intrusion detectors (classifiers) is fine-tuned according to the corresponding

features. This approach might also be very effective, since the four light-weight classifiers can be operated in parallel.

In order to perform the experiment 2), we added normal traffic into each attack class to get four data sets: KDD99-normal&DoS, KDD99-normal&Probe, KDD99-normal&U2R and KDD99-normal&R2L. With each data set, we ran three feature selection algorithms: our new CFS-based method, the best-first CFS-based and the genetic algorithm CFS-based methods. The numbers of selected features and their identifications are given in Tables II and III, respectively. We then applied the C4.5 and the BayesNet machine learning algorithms on each original full set as well as each newly obtained data set that includes only the features obtained from the feature selection algorithms. We applied 5-fold cross-validation on each data set. The classification accuracies are reported in Table IV.

Our new CFS-based method was compared with the best-first-CFS and genetic-algorithm-CFS methods regarding the number of selected features and regarding the classification accuracies of 5-fold cross-validation of BayesNet and C4.5 learning algorithms. Weka tool [13] was used for obtaining the results. In order to solve the $M01LP$ problem, we used TOMLAB tool [12].

B. Experimental Results

Table II shows the number of features selected by using our approach and those selected by using the best-first and GA search strategies. The identification of selected features is given in Table III (for feature names, see Appendix A). Table IV summarizes the classification accuracies of the BayesNet and the C4.5 performed on four data sets (see above).

It can be observed from Table II that our CFS-based approach selects the smallest number of relevant features in comparison with the feature sets selected by the best-first and GA search strategies. Especially in some cases, our new method compresses the full set of features extremely. For example, only one feature was selected out of 41 features of the KDD99-normal&U2R data set.

In the Table IV, it can be observed that with our approach the average classification accuracies are slightly different from the ones obtained by using the best-first search or the genetic algorithm. The absolute difference between them does not overcome 0.69%. In the case of the C4.5 classifier, we got better performance. Even though the gain of classification accuracy is not very high compared to other methods, the overall gain of the feature selection and classification procedure lies in significantly improved efficiency and in obtaining good classification results with a reduced number of relevant features.

Therefore, based on all the experiments, we can say that in general our new method outperforms the best-first-CFS and genetic-algorithm-CFS methods by removing much more redundant features and still keeping the classification accuracies or even getting better performances. Thus it can be used to find optimal subsets of relevant features by means of the CFS measure for intrusion detection systems.

V. Conclusion

We proposed a new method to get the globally optimal subset of relevant features by means of the correlation feature selection (CFS) measure. We transformed the CFS optimization problem into a polynomial mixed $0-1$ fractional programming $(P01FP)$ problem. From this $P01FP$, we then applied an improved Chang's method to get a mixed $0-1$ linear programming $(M01LP)$ problem with linear dependence of the numbers of constraints and variables on the number of features in the full set. We used the branch-and-bound algorithm to solve that $M01LP$. Experimental results show that our approach outperforms the best-first-CFS and genetic-algorithm-CFS methods by removing much more redundant features and still keeping the classification accuracies or even getting better performances.

Appendix A
Names and identifications (ID) of selected features

Here we enumerate the features from the original KDD CUP'99 data set used in Table III.

ID	Names
5	src_bytes
6	dst_byte
10	hot
12	logged_in
14	root_shell
22	is_guest_login
29	same_srv_rate
37	dst_host_srv_diff_host_rate
41	dst_host_srv_rerror_rate

References

[1] C. T. Chang, "An efficient linearization approach for mixed integer problems", European Journal of Operational Research, vol. 123, pp. 652-659, 2000.

[2] C. T. Chang, "On the polynomial mixed 0-1 fractional programming problems", European Journal of Operational Research, vol. 131, pp. 224-227, 2001.

[3] Y. Chen, Y. Li, X. Q. Cheng and L. Guo, "Survey and Taxonomy of Feature Selection Algorithms in Intrusion Detection Systems", in Proceedings of Inscrypt 2006, LNCS 4318, pp. 153-167, 2006.

[4] G. Di Crescenzo, A. Ghosh, and R. Talpade, "Towards a theory of intrusion detection", in Proceedings of ESORICS 2005, LNCS 3679, pp. 267-286, 2005.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, John Wiley& Sons, USA, 2nd edition, 2001.

[6] G. Gu, P. Fogla, D. Dagon, W. Lee, and B. Skoric, "Towards an information-theoretic framework for analyzing intrusion detection systems", in Proceedings of ESORICS 2006, LNCS 4189, pp. 527-546, 2006.

[7] I. Guyon, S. Gunn, M. Nikravesh and L. A. Zadeh, Feature Extraction: Foundations and Applications, Series Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer, 2006.

[8] M. Hall, Correlation Based Feature Selection for Machine Learning, Doctoral Dissertation, University of Waikato, Department of Computer Science, 1999.

[9] KDD Cup 1999 data set. http://www.sigkdd.org/kddcup/index.php?section=1999&method=data.

TABLE II
NUMBER OF SELECTED FEATURES (GA: GENETIC ALGORITHM)

Data Set	Full-set	Our-method	Best-first	GA
KDD99-normal&Dos	41	3	6	11
KDD99-normal&Probe	41	6	7	17
KDD99-normal&U2R	41	1	4	8
KDD99-normal&R2L	41	2	5	8

TABLE III
IDENTIFICATIONS OF SELECTED FEATURES (FOR FEATURE NAMES, SEE APPENDIX A)

Data Set	Identifications
KDD99-normal&Dos	5, 6, 12
KDD99-normal&Probe	5, 6, 12, 29, 37, 41
KDD99-normal&U2R	14
KDD99-normal&R2L	10, 22

TABLE IV
CLASSIFICATION ACCURACIES OF C4.5 AND BAYESNET PERFORMED ON KDD CUP'99 DATA SET

Data Set	C4.5				BayesNet			
	Full-Set	Our-method	Best-First	GA	Full-Set	Ours	Best-First	GA
KDD99-normal&DoS	97.80	98.89	96.65	96.09	99.99	98.87	99.09	99.72
KDD99-normal&Probe	99.98	99.70	99.71	99.89	98.96	97.63	97.65	99.19
KDD99-normal&U2R	99.97	99.96	99.97	99.95	99.85	99.95	99.97	99.93
KDD99-normal&R2L	98.70	99.11	99.01	98.86	99.33	98.81	98.95	99.28
Average	**99.11**	**99.41**	**98.84**	**98.69**	**99.53**	**98.82**	**98.91**	**99.52**

[10] H. Liu, H. Motoda, Computational Methods of Feature Selection, Boca Raton: Chapman & Hall/CRC, 2008.

[11] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[12] TOMLAB, The optimization environment in MATLAB. http://tomopt.com/tomlab/.

[13] Weka, the data mining software in Java. http://en.wikipedia.org/wiki/Weka_(machine_learning).

Control de acceso interoperable para la mejora en la cooperación entre grupos de emergencias

Carles Martínez-García*, Abraham Martín-Campillo*, Guillermo Navarro-Arribas†, Ramon Martí* and Joan Borrell*
* Department of Information and Communications Engineering (dEIC)
Universitat Autònoma de Barcelona
08193 Bellaterra, Spain
Email: {carlos.martinez, abraham.martin, ramon.marti.escale, joan.borrell}@uab.cat
†IIIA, Institut d'Investigació en Intel·ligència Artificial -
CSIC, Consejo Superior de Investigaciones Científicas,
Campus UAB s/n, 08193 Bellaterra, Spain
Email: guille@iiia.csic.es

***Resumen*—La cooperación entre los diferentes grupos que intervienen en el control y gestión de emergencias no es algo trivial. Actualmente esta cooperación se realiza de manera física, con comunicaciones verbales e informes que hacen que no sea todo lo rápida que cabe esperar. De la mano de la informatización de los sistemas de soporte a la emergencia, nace la necesidad de compartir la información recogida por los diferentes grupos, con el objetivo de aunar esfuerzos. En el presente documento, afrontamos la interoperabilidad desde la vertiente de control de acceso. Para ello, proponemos un mecanismo de conversión de atributos que permite a los miembros de cada grupo de la emergencia actuar en calidad de usuarios en los sistemas del resto de grupos. Políticas de conversión establecen relaciones entre puestos jerárquicos equivalentes de cada grupo. El mecanismo contempla el error intrínseco en la conversión así como gestiona su transitividad permitiendo una más ágil, sencilla y segura cooperación entre todos los actores en una emergencia.**

I. Introducción

A partir del instante en que se produce una emergencia, la máxima prioridad es la atención a las víctimas. Para ello, diferentes grupos o equipos -como podrían ser equipos médicos, el cuerpo de bomberos, protección civil o la policía local- se desplazan hasta la zona de la catástrofe con el objetivo de atender a los directamente damnificados por ésta y gestionar y controlar la situación. Actuaciones como, por ejemplo, la rápida identificación del estado de las víctimas -incluyendo su localización espacial y su nivel de gravedad-, asegurar el área de la catástrofe y la gestión del traslado de las víctimas a centros hospitalarios son desplegadas por diferentes grupos.

Dada la ausencia de un único sistema de soporte y gestión de la emergencia, cada grupo de rescate establece su propio protocolo de actuación. Sin embargo, con el objetivo de aunar esfuerzos para una mejor gestión y control de las catástrofes, se crea la necesidad de una integración de la información relacionada con la emergencia. De esta manera, el equipo médico podría acceder, por ejemplo, a datos pertenecientes a los bomberos y relacionados con la extracción de víctimas o información, descrita por la policía, sobre el alcance real de la emergencia para poder precisar el numero de efectivos que se van a necesitar. A pesar de que actualmente se establece un punto de coordinación donde se sitúan los coordinadores de cada grupo, es precisamente aquí dónde se encuentran los cuellos de botella de los flujos de información entre el personal de campo.

El uso de sistemas informatizados de soporte a la emergencia debería facilitar la interoperabilidad entre grupos. Sin embargo, ésta requiere esfuerzos que engloban varia disciplinas como el diseño de interfaces, la interconexión de redes o la propia seguridad. En términos exclusivos de autorización, el acceso a la información compartida se debe regular para evitar comprometer datos de alta sensibilidad. Partiendo de la base de que cada grupo de rescate presenta una estructura jerárquica interna que regula el nivel de acceso de sus propios integrantes, el problema de la interoperabilidad no resulta trivial. La información es sensible, máxime cuando se refiera a víctimas, y por ello debe ser tratada de manera apropiada. En consecuencia, es necesario garantizar que el acceso a los datos se haga solamente por aquellas personas autorizadas dada su jerarquía.

En el presente documento proponemos un mecanismo para convalidar, de la manera más aproximada posible, las credenciales que posee un sujeto dentro del grupo al que pertenece por credenciales de un grupo cooperante. Esto permite equiparar el nivel de acceso de los integrantes de cada grupo sin necesidad de cambiar las políticas de control de acceso de cada grupo ni su propio mecanismo de autorización. Dado que en general no será posible encontrar una relación directa entre el nivel de acceso en diferentes grupos debido a su heterogeneidad, el mecanismo propuesto contempla la tasa de error en la convalidación. Un umbral de seguridad, ajustable al tempo de la emergencia, establecerá la cota máxima aceptable del error. Finalmente, cabe destacar que el mecanismo es capaz de convalidar atributos de forma transitiva con el objetivo de habilitar la interoperabilidad entre grupos aun cuando no exista una política de conversión explícitamente definida.

El resto del artículo se organiza de la manera siguiente: En la Sección II se muestra un estado del arte de la propuesta centrándose en las aplicaciones para la gestión y seguridad de

emergencias. Posteriormente en la Sección III se describe la propuesta, mostrando su funcionamiento en la Sección IV a través de un ejemplo. En la Sección V se expone una breve descripción de la implementación realizada y por último la Sección VI concluye el artículo.

II. Estado del arte

II-A. Aplicaciones para la gestión de la emergencia

Dentro de una emergencia de gran abasto, suelen actuar varios equipos de rescate, atención a la víctimas y gestión y control de la catástrofe. Cabe remarcar que cada grupo utiliza sus propios recursos de coordinación que, actualmente, no comparten con los demás. Es posible que el lector pueda llegar a pensar que los equipos involucrados en la emergencia no utilizan ningún tipo de sistema informatizado y que esto es algo futurible, pero no es del todo cierto. Existen proyectos como el propuesto por la Generalitat de Catalunya [4], que prevé dotar a todos los equipos de bomberos de localizador GPS y cámaras para poder ver la emergencia en tiempo real. Además, trabajos existentes [11] [3] [10] [13] instan a la informatización de algunos de estos equipos.

II-B. Seguridad

A nuestro entender, no existen publicaciones que describan mecanismos de control de acceso específicos para aplicaciones destinadas al control y gestión de grandes emergencias. Sin embargo, esquemas clásicos como RBAC [5] pueden ser aplicables a estos entornos. Cabe decir que el control de acceso en estas situaciones es crucial. Sin embargo, el compromiso inherente entre seguridad y flexibilidad adquiere una nueva dimensión en este tipo de entornos. Mientras proteger la confidencialidad de los datos es fundamental, facilitar su acceso puede ayudar a salvar vidas. Propuestas como el concepto de *override* [15] otorgan al usuario cierto poder de decisión aun cuando sus peticiones de acceso han sido denegadas. De forma mas radical, se pueden encontrar propuestas como [14] cuyo objetivo es proponer un sistema de control de acceso disuasivo en vez de restrictivo. A grandes rasgos, el control de acceso disuasivo permite por defecto ejecutar todas las acciones auditando a posteriori las posibles violaciones de la política.

La cooperación entre grupos guarda una semejanza inherente al control de acceso en entornos multidominio. Actualmente existen algunas propuestas para solucionar problemas de control de acceso en este tipo de entornos. Por un lado, encontramos la interoperabilidad a nivel de política donde la generación de políticas de control de acceso permite compartir recursos entre dominios. Dos alternativas de esta solución [8] son los llamados *dominios libremente asociados* y los *dominios federados*. Los primeros se basan en la generación de políticas de control de acceso locales [17] que permiten a cada dominio dar autorización a usuarios de otros dominios. En los segundos, un sólo dominio máster tiene políticas de control de acceso global [16] que regulan todas las acciones que se realizan en el escenario completo.

Por otro lado, la interoperabilidad a nivel de atributos se basa en encontrar una relación entre los atributos de diferentes dominios. A este nivel, generalmente la interoperabilidad parte de la premisa de que se podrán encontrar atributos completamente equivalentes entre los diferentes dominios, cuando a la práctica es bastante improbable. Bajo esta línea, se hallan propuestas como un mecanismo de conversión de atributos a través de políticas [9] o el uso de ontologías [19] para especificar información basada en el contexto bajo el cual se pueden definir relaciones entre roles de los diferentes dominios que usan RBAC. Otra propuesta relacionada [12] presenta un sistema de conversión cuantificada aportando un nivel de flexibilidad a la hora de establecer relaciones entre atributos que no guardan una equivalencia total. Sin embargo, todas las propuestas citadas pecan de falta de escalabilidad, pues cada dominio debería poseer tantas políticas de conversión como dominios haya en el escenario, menos uno: la que establecería una conversión hacia sí mismo.

De una forma mas tangencial, pero estrechamente relacionada con la conversión de atributos encontramos propuestas como la presentada por Foley [6] donde se proponen medidas de similitud entre atributos de un mismo dominio con tal de flexibilizar el proceso de autorización. De esta manera, el sistema debería autorizar acciones si el sujeto presenta un grupo de credenciales suficientemente cercanos a los credenciales requeridos en la política.

III. Propuesta

En esta sección describimos la propuesta del mecanismo que habilita la interoperabilidad de control de acceso en entornos de emergencias y que, además, permite definir de una forma ágil la frontera inherente entre seguridad y usabilidad en cada momento. El mecanismo que proponemos parte de la premisa de que generalmente no será posible encontrar atributos en diferentes dominios con una equivalencia total y, además, ayuda a la escalabilidad habilitando conversiones de atributos de forma transitiva.

III-A. Conceptos previos

Previa a la descripción del mecanismo de interoperabilidad, se hace necesaria una descripción, en términos de control de acceso, de la estructura administrativa de los distintos grupos que pretenden cooperar. Dada la heterogeneidad que éstos ofrecen, fruto de su independencia, presentamos una visión lo suficientemente genérica que engloba gran parte de los mecanismos de control de acceso con los que nos encontramos en la actualidad, desde sistemas de control de acceso basados en identidad hasta RBAC.

Consideramos que cada grupo representa un dominio de seguridad. Cada dominio posee, pues, un conjunto de *sujetos* dentro de él. Los sujetos son entidades activas que aplican un conjunto de *acciones* definidas sobre el conjunto de *objetos* del sistema. Los sujetos del sistema se caracterizan mediante *atributos*. Un atributo es cualquier información relacionada con los sujetos, a través de una autoridad central, y que puede ser considerada durante la toma de decisiones de acceso.

Un mecanismo de control de acceso regula la ejecución de acciones por parte de los sujetos sobre los objetos. Diversas políticas expresan los atributos cuya posesión, por parte de los sujetos, habilita la ejecución de dichas acciones.

III-B. Habilitando la interoperabilidad

El mecanismo de interoperabilidad en términos de control de acceso que proponemos se fundamenta a nivel de atributo. La idea principal es establecer una relación entre los atributos de dos dominios distintos, de tal manera que, por ejemplo, el jefe de bomberos se pueda equiparar al jefe del personal médico de rescate. Para ello se definen políticas de conversión de atributos que establecen una relación directa entre los atributos de dos dominios. De esta manera, los atributos relacionados con un sujeto dentro de un dominio en el cual el sujeto no es autóctono se calculan directamente teniendo en cuenta los atributos del usuario en su dominio de origen. Una vez computados los nuevos atributos del usuario, éste puede actuar en el sistema como un usuario mas. Por consiguiente, la interoperabilidad no requiere la modificación ni de las políticas de control de acceso de cada dominio ni de los mecanismos de seguridad.

Debido a la heterogeneidad en la definición de los atributos por parte de los distintos dominios, como norma general no será posible definir una relación de equivalencia total entre dos atributos. Dicho de otra manera, difícilmente habrá dos atributos en dominios diferentes que garanticen un nivel de acceso idéntico. El nivel de similitud entre dos atributos denota el error intrínseco en la conversión. Para tratar el error de una forma natural proponemos la siguiente definición de políticas de conversión:

Definición 1. *Las* políticas de conversión de atributos *entre dos dominios A y B, denotadas como C_{AB}, se definen como una relación difusa entre los atributos de A y B, tal que: $C_{AB} : A \times B \rightarrow [0,1]$, donde 0 significa que el grado de equivalencia entre dos atributos es nulo y 1 representa el máximo grado de similitud.*

Intuitivamente, si $C_{AB}(a,b) = 0{,}8$, implica que el atributo a definido en el dominio A se considera $0{,}8$ equivalente al atributo b definido en B por la propia política C_{AB}.

III-C. Interoperabilidad y transitividad

La definición de políticas de conversión de atributos habilita la interoperabilidad entre dos grupos en el escenario de la emergencia. Por el contrario, se puede dar la situación de que no exista ninguna política de conversión de atributos directa entre dos grupos distintos que pretenden cooperar. Intuitivamente, se puede considerar que si existe una política de conversión entre el dominio A y el dominio B, y otra política entre el dominio B y el dominio C, los atributos del dominio A se pueden convertir en atributos del dominio C a través del dominio B. Sin embargo, cabe esperar la propagación de los grados de similitud durante la transitividad. Ésta no solo beneficia al sistema en términos de escalabilidad, fruto de la reducción del número de políticas necesarias para convertir los atributos de un dominio hacia el resto, sino que aporta agilidad al sistema a la hora de efectuar colaboraciones en escenarios donde el tiempo de respuesta es crucial.

A fin de que la transitividad se pueda ejecutar de manera eficiente proponemos un mecanismo de composición de políticas de conversión C_{AB} y C_{BC}. Dadas dos políticas de conversión, definimos su composición como:

$$C_{AC} : C_{AB} \circ C_{BC}$$

Donde el operador de composición [18] se define como:

$$C_{AB} \circ C_{BC} = \max_{b \in B} \min(C_{AB}(a,b), C_{AB}(b,c))$$

Bajo la presencia de diferentes caminos de conversión entre dos dominios -es decir, los atributos del dominio A se pueden convertir en atributos del dominio B a través del dominio C o del dominio D-, el mecanismo de conversión procede de la siguiente manera: Primero se encuentran todos los caminos de conversión acíclicos entre el dominio origen y destino. Acto seguido se calcula la composición de las políticas de conversión que forman parte de cada camino. Finalmente, se computa la política resultante como la unión de las políticas compuestas de cada camino. La unión de n políticas se computa escogiendo el valor mayor, posición a posición, de cada una de las n políticas. Si no fuese viable, en términos computacionales o debido a restricciones de tipo temporal, encontrar todos los caminos de conversión entre el dominio origen y destino, soluciones sub-optimas pueden ser alcanzadas restringiendo el numero de caminos usados en la solución.

Una vez computada la relación transitiva entre el dominio origen y destino, la conversión de atributos puede efectuarse sobre dicha relación. Nótese que la función no solo establece la transitividad en la conversión sino que además propaga el error intrínseco de forma natural.

III-D. El umbral de seguridad

La interoperabilidad a nivel de atributo entraña que el mecanismo de conversión realice asignaciones automáticas de atributos a usuarios provenientes de dominios externos. Debido a que es el propio dominio destino quien asume el riesgo en la conversión, éste se reserva las acciones de comprobación necesarias previa asignación de un atributo a un sujeto.

El nivel de similitud entre dos atributos se debe tener en cuenta previamente a la conversión. Un grado de similitud demasiado bajo puede no justificar la conversión de un atributo. Para ello definimos el umbral de seguridad δ como:

$$\delta \rightarrow [0,1]$$

El umbral de seguridad δ, presente en cada dominio, establece el grado de similitud mínimo por el cual un atributo puede ser convertido. De esta manera, si la magnitud de conversión es superior o igual a δ, la asignación del atributo convertido se hace efectiva. Dicho de otra manera, dado un atributo a definido en el dominio A y un atributo b definido en el dominio B:

$$a \sim b \; sii \; C_{AB}(a,b) \geq \delta$$

Cabe destacar que el umbral δ representa la frontera entre flexibilidad y seguridad en el mecanismo propuesto y que éste debe ser debidamente ajustado de manera acorde al tempo de la emergencia, siendo mas permisivo en las fases que así lo requieran.

IV. Ejemplo

En esta sección se expone un breve ejemplo en pro de la compresión del mecanismo de interoperabilidad.

IV-A. Escenario de la emergencia

Suponga un escenario de emergencia en el que intervienen tres grupos. El primer grupo es el grupo de bomberos, que presenta la estructura jerárquica mostrada en la Figura 1. El segundo grupo es el formado por el Servicio de Emergencias Médicas estructurado tal y como se muestra en la Figura 2. Por último, el tercer grupo, está formado por la policía local del municipio donde ha ocurrido el accidente y presenta la estructura mostrada en la Figura 3.

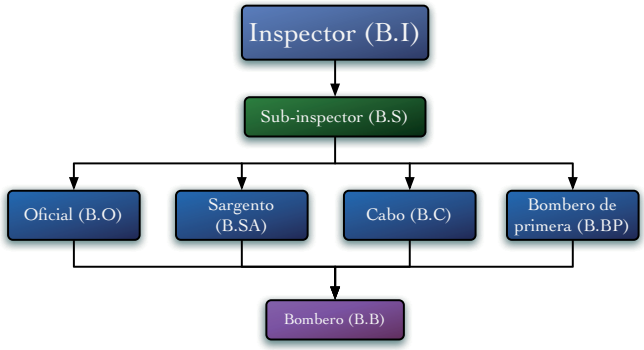


Figura 1. Organigrama del cuerpo de bomberos [7].

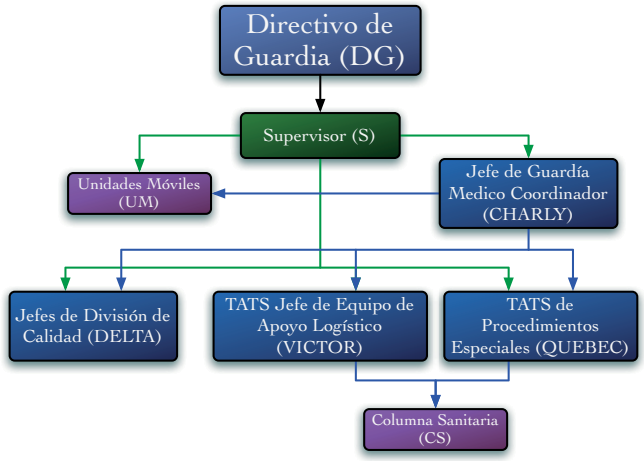


Figura 2. Organigrama del cuerpo médico [2] simplificado.

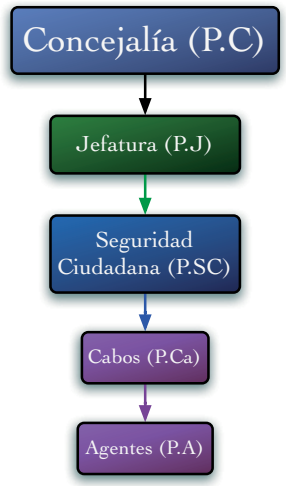


Figura 3. Organigrama del cuerpo de policía [1].

Cuadro I
Política de conversión de atributos entre el cuerpo de bomberos y el equipo de emergencias medicas

	DG	S	C	V	Q	UM	CS
B.I	0,8	0,9	0	0	0	0	0
B.S	0	0,7	0,9	0	0	0	0
B.O	0	0	0	1	0,5	0	0
B.C	0	0	0	0,9	0,8	0	0
B.BP	0	0	0	0	0	1	0,8
B.B	0	0	0	0	0	0,8	1

Con el objetivo de agilizar la atención a las víctimas y la gestión y control de la emergencia, los tres grupos implicados en la catástrofe deciden cooperar. Para ello, habilitan el acceso a recursos compartidos entre los diferentes equipos. Estos recursos contienen información sobre la clasificación y localización de las víctimas o el alcance de la emergencia, entre otros datos de importancia. En el Cuadro I se muestra la política de conversión de atributos entre el cuerpo de bomberos y el equipo de emergencias médicas. En el Cuadro II, la política de conversión de atributos entre el equipo de emergencias médicas y el cuerpo de policía local.

Mediante las políticas de conversión, se considera que el cargo de *bombero* (B.B) es similar en un 80 % al cargo de *personal de unidad móvil* (UM) y 100 % similar al puesto *personal de columna sanitaria* (CS). Siempre y cuando este

Cuadro II
Política de conversión de atributos entre el equipo de emergencias medicas y el cuerpo de policía local

	P.C	P.J	P.SC	P.Ca	P.A
DG	1	0	0	0	0
S	0	1	0	0	0
C	0	0	1	0	0
V	0	0	0	0,9	0
Q	0	0	0	0,8	0
UM	0	0	0	0	1
CS	0	0	0	0	1

Cuadro III
POLÍTICA DE CONVERSIÓN DE ATRIBUTOS ENTRE EL CUERPO DE BOMBEROS Y EL CUERPO DE POLICÍA LOCAL, FRUTO DE LA COMPOSICIÓN DE LAS POLÍTICAS MOSTRADAS EN LAS CUADROS I Y II.

	P.C	P.J	P.SC	P.Ca	P.A
B.I	0,8	0,9	0	0	0
B.S	0	0,7	0,9	0	0
B.O	0	0	0	0,9	0
B.C	0	0	0	0,9	0
B.BP	0	0	0	0	1
B.B	0	0	0	0	1

valor sea igual o mayor que el umbral de seguridad δ, la conversión de los atributos se considera efectiva equiparando el nivel de acceso de ambos atributos.

Si bien es cierto que no existe una relación directa entre los atributos del cuerpo de bomberos y los atributos del cuerpo de policía local, no es menos cierto que existe una relación implícita, representada en el Cuadro III, fruto de la composición de las políticas de conversión mostradas en los Cuadros I y II.

Mediante la composición de las políticas de conversión, el atributo *bombero* (B.B) guarda una similitud del 100 % con el atributo *agente* (P.A). Cabe destacar que, en este caso, la conversión de atributos se considerará siempre efectiva, debido a que el grado de similitud nunca estará por debajo del umbral de seguridad δ.

V. IMPLEMENTACIÓN

Se ha desarrollado un prototipo que permite definir los diferentes dominios de seguridad en el escenario de la emergencia así como las políticas de conversión de atributos entre ellos. El prototipo no solo permite la conversión de atributos a través de políticas explícitamente definidas, sino que es capaz de convertir atributos a través de la composición de relaciones.

Una interfaz gráfica (ver Figura 4) permite definir los diferentes dominios de seguridad, así como sus atributos. Posteriormente, el prototipo permite definir las políticas de conversión de atributos entre los dominios implicados. Una vez establecidas las políticas de conversión (ver Figura 5), se permite lanzar conversiones de atributos encargándose de la composición de políticas si fuese necesario.

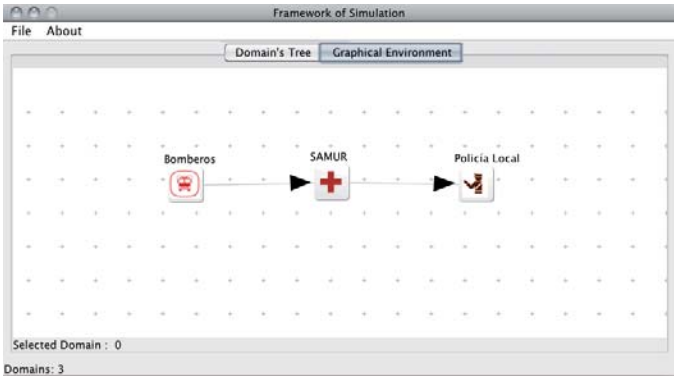


Figura 4. Interfaz que muestra las relaciones entre atributos del ejemplo

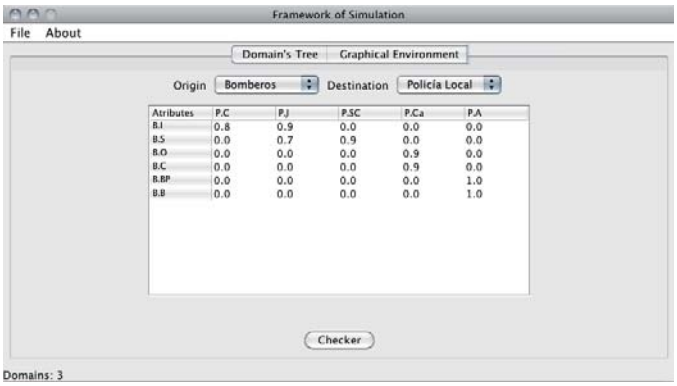


Figura 5. Relación entre los atributos de los bomberos y la policía.

VI. CONCLUSIONES

La cooperación entre los diferentes equipos que trabajan de forma concurrente en escenarios de emergencia es esencial para aunar los esfuerzos en la atención a las víctimas, el control y la gestión de la emergencia. Sin embargo, la heterogeneidad que presentan los grupos a nivel de estructura, jerarquía y sistemas de soporte a la emergencia dificulta enormemente las tareas de colaboración. En el presente documento hemos propuesto un mecanismo de conversión de atributos que, dados los atributos otorgados a un sujeto en el grupo de rescate al que pertenece, es capaz de convalidarlos en atributos del resto de grupos inmersos en el rescate. De esta manera, los usuarios son capaces de actuar de forma natural dentro del resto de grupos cooperantes.

El mecanismo propuesto contempla el error intrínseco en la convalidación y es capaz de propagarlo en la transitividad de las políticas de conversión. De esta manera, es posible mantener acotado el error que cada sistema está dispuesto a tolerar en cada momento. Además, la transitividad en las políticas de conversión es tratada de forma natural ayudando a la escalabilidad del mecanismo fruto de la reducción del numero de políticas explícitamente definidas en el escenario.

Se ha desarrollado un prototipo que, previa definición de varios dominios en un escenario, permite lanzar conversiones de atributos encargándose de computar las políticas transitivas si ello fuera necesario. Como trabajo futuro, se propone desarrollar el mecanismo en forma de *middleware* que permita la interoperabilidad de los diferentes equipos implicados en la emergencia siendo, a su vez, lo menos intrusivo posible.

VII. AGRADECIMIENTOS

Este trabajo está respaldado por el Ministerio de Educación y Ciencia (FPU AP2008-03149) el MICCINN (proyectos eAEGIS TSI2007-65406-C03-02, y ARES – CONSOLIDER INGENIO 2010 CSD2007-00004), por el Departament d'Innovació, Universitats i Empresa (2009SGR1224) y por la Universitat Autònoma de Barcelona (PIF 472-01-1/07). G. Navarro-Arribas disfruta de una beca Juan de la Cierva (JCI-2008-3162) del MICINN.

Por último, agradecer de forma especial a Fco. Javier Llorente Palacio por su dedicación en el diseño e implementación del prototipo.

REFERENCIAS

[1] Organigrama de la policía local de molina de segura (murcia). http://www.molinadesegura.es/policialocal/porganigrama.htm.

[2] Comunidad de Madrid. Samur protección civil - organigrama operativo. http://www.munimadrid.es/UnidadesDescentralizadas/Emergencias/Samur-PCivil/Samur/Ficheros/Organizacion/organigrama_operativo.pdf.

[3] B. Demchak, T. C. Chan, W. G. Griswold, and L. A. Lenert. Situational awareness during mass-casualty events: command and control. *AMIA Annu Symp Proc*, page 905, 2006.

[4] El País. La comisión del fuego de horta se salda con 66 propuestas para mejorar la tarea de los bomberos, March 2010.

[5] D. Ferraiolo, R. Sandhu, S. Gavrila, and D. Kuhn. Proposed nist standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)*, Jan. 2001.

[6] S. Foley. Supporting imprecise delegation in KeyNote using similarity measures. In *Sixth Nordic Workshop on Secure IT Systems*, pages 101–119, 2001.

[7] Generalitat de Catalunya. Escala del cos de bombers de la generalitat. http://www.gencat.cat/interior/esc/docs/esc_galons.htm.

[8] J. B. D. Joshi, R. Bhatti, E. Bertino, and A. Ghafoor. Access-control language for multidomain environments. *IEEE Internet Computing*, 8(6):40–50, 2004.

[9] G. López, O. Cánovas, and A. Gómez-Skarmeta. Use of XACML policies for a network access control service. In *4th International Workshop for Applied PKI, IWAP 05*, pages 111–122, Sept. 2005.

[10] K. Luyten, F. Winters, K. Coninx, D. Naudts, and I. Moerman. A situation-aware mobile system to support fire brigades in emergency situations. *On the Move to Meaningful Internet Systems 2006: Otm 2006 Workshops, Pt 2, Proceedings*, 4278:1966–1975, 2006.

[11] R. Martí, S. Robles, A. Martín-Campillo, and J. Cucurull. Providing early resource allocation during emergencies: the mobile triage tag. *Journal of Network and Computer Applications*, 32(6):1167–1182, November 2009.

[12] C. Martínez-García, G. Navarro-Arribas, J. Borrell, and A. Martín-Campillo. Sistema impreciso de control de acceso basado en la conversión cuantificada de atributos para escenarios de interoperabilidad. In L. Hernández and Ángel Martín, editors, *Actas de la X RECSI*, pages 357–367, Salamanca, September 2008. Universidad de Salamanca.

[13] S. McGrath, E. Grigg, S. Wendelken, G. Blike, M. D. Rosa, A. Fiske, and R. Gray. Artemis: A vision for remote triage and emergency management information integration. pages–. Dartmouth University, November 2003.

[14] D. Povey. Optimistic security: a new access control paradigm. In *NSPW '99: Proceedings of the 1999 workshop on New security paradigms*, pages 40–45, New York, NY, USA, 2000. ACM.

[15] M. Sergot, B. Sadighi, and E. Rissanen. Towards a mechanism for discretionary overriding of access control: position paper. In *Proceedings of Proceedings of the twelfth international workshop on security protocols*, page 9, Cambridge, UK, 2004.

[16] B. Shafiq, J. Joshi, E. Bertino, , and A. Ghafoor. Optimal secure interoperation in a multi-domain environment employing RBAC policies. Technical Report 2003-24, CERIAS, Purdue University, 2003.

[17] B. Shafiq, J. Joshi, E. Bertino, and A. Ghafoor. Secure interoperation in a multidomain environment employing RBAC policies. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1557–1577, 2005.

[18] V. Torra and Y. Narukawa. *Modeling Decisions: Information Fusion and Aggregation Operators*. Springer, 2007.

[19] D. Wu, X. Chen, J. Lin, and M. Zhu. Ontology-based RBAC specification for interoperation in distributed environment. In *The Semantic Web - ASWC 2006, First Asian Semantic Web Conference*, pages 179–190, Sept. 2006.

Modelo criptobiométrico de liberación de clave basado en firmas en el aire

J. Guerra Casanova, C. Sánchez Ávila, A. de Santos Sierra, G. Bailador del Pozo, V. Jara Vera
Centro de Domótica Integral (CeDInt-UPM) Universidad Politécnica de Madrid
Campus de Montegancedo, 28223 Pozuelo de Alarcón, Madrid
Email:{jguerra, csa, alberto, gbailador, vjara}@cedint.upm.es

***Resumen*—En este artículo se propone una técnica biométrica basada en firmas en el aire. Los usuarios se autentican realizando una firma en el aire con un dispositivo móvil que incluya un acelerómetro. Para ello, se propone un sistema en el que todas las operaciones necesarias para el proceso de autenticación se llevan a cabo en el propio dispositivo, lo cual permite proponer un modelo criptobiométrico de liberación de clave. En este modelo, hay una clave criptográfica almacenada en el teléfono móvil y vinculada a la identidad del usuario propietario del mismo. El usuario puede liberar la clave mediante la realización de su firma en el aire, y utilizarla como clave criptográfica asociada a su identidad, lo cual le puede permitir realizar distintas operaciones en Internet que necesitan autenticación de la identidad de la persona. Este artículo trata de validar la técnica biométrica en la que se basa el modelo criptobiométrico, para ello, se ha creado una base de datos de firmas en el aire de 34 usuarios. Además, tres falsificadores han tratado de imitar cada una de las firmas originales a partir de grabaciones de video. A partir del análisis de las muestras originales y falsificaciones se ha obtenido una tasa de Equal Error Rate de 2.5 %, consumiendo en todo el proceso de autenticación menos de dos segundos.**

I. Introducción

Hoy en día se puede acceder a aplicaciones en Internet que pueden necesitar autenticación desde la mayoría de dispositivos móviles. Mirar el saldo de una cuenta corriente, comprar un producto en una tienda online o realizar ciertas operaciones en sitios web seguros son solo algunos ejemplos de operaciones que se pueden realizar desde un móvil con acceso a Internet y donde es importante que el usuario sea quien dice ser.

En este contexto móvil, la seguridad suele dejarse en manos de contraseñas o códigos PIN que se supone que sólo el usuario sabe. Pero esto esconde un gran riesgo ya que las contraseñas pueden ser robadas o adivinadas comprometiendo la seguridad del sistema.

El campo de la criptografía coincide con esta problemática, ya que la fortaleza de los protocolos criptográficos reside en la facilidad para averiguar la clave del usuario. Una vez que la clave está comprometida, el sistema no puede considerarse seguro.

La utilización de técnicas biométricas permite solucionar estos problemas. Por un lado, el usuario no puede olvidarse de su clave, puesto que él mismo es la clave. Por otro lado, si la técnica biométrica es suficientemente distintiva, ningún usuario va a poder autenticarse en el sistema como si fuera otro, manteniendo la clave del usuario original completamente segura.

En la actualidad existen varias investigaciones que tratan de unir las técnicas clásicas de biometría en escenarios móviles. Por ejemplo, en [1] se presenta un trabajo basado en reconocimiento de iris mediante cámaras en teléfonos móviles, en [2] estas cámaras se utilizan para realizar reconocimiento facial, en [3] la autenticación se basa en las características de la voz extraidas al hablar por teléfono y en [4] se hace una evaluación de todas las técnicas de reconocimiento biométrico anteriores en teléfonos móviles.

Asímismo, existen una gran cantidad de trabajos que tratan de esquemas de generación o liberación de claves criptográficas a partir de patrones biométricos [5] en una rama de investigación que está naciendo y que se denomina criptobiometría.

En este artículo se propone una nueva técnica biometrica basada en la realización de la firma en el aire con la mano sujetando un teléfono móvil que integre un acelerómetro. A partir de esta técnica se va a proponer un modelo criptobiométrico muy sencillo, en el que al autenticarse un usuario se libera una clave criptográfica almacenada en el propio dispositivo, de tal manera que únicamente el usuario que se haya registrado en su teléfono móvil va a poder acceder a realizar ciertas transacciones seguras con esa clave privada almacenada. Este tipo de modelo criptobiométrico es el más sencillo y puede valer de estudio previo para otros modelos más complejos con los que se pueden obtener mejores resultados.

Para estudiar la validez de la técnica, se ha realizado una base de datos de 34 usuarios que han realizado una firma o gesto identificativo en el aire con un iPhone (que integra un acelerómetro). Además, se ha propuesto un método matemático de análisis de las señales basado en Programación dinámica, así como distintas maneras de fusionar la información de los acelerómetros en cada eje.

Este artículo se divide en la siguientes secciones. En primer lugar, en la Sección II se detallan las características de la técnica criptobiométrica propuesta en este articulo. A continuación, en la Sección III se describe el método matemático de análisis de las señales extraídas de las firmas en el aire correspondientes a esta técnica. Más adelante, en la Sección IV se presenta cómo ha sido la creación de la base de datos de firmas en el aire que se ha utilizado para dar soporte a los experimentos del artículo. La Sección V incluye una explicación del trabajo experimental que se ha llevado a cabo,

así como el tiempo y las tasas de error que se han obtenido. Finalmente, en la Sección VI se presentan las conclusiones de este trabajo junto a unas líneas futuras para seguir investigando en el tema.

II. Descripción de la técnica criptobiométrica propuesta.

Esta técnica se basa en la realización de una firma en el aire con la mano sujetando un teléfono móvil. Para ello, es necesario que el teléfono móvil integre un acelerómetro, con el que se va a extraer la información de las aceleraciones en el eje X, Y y Z de la firma en el aire del usuario. Actualmente que la mayoría de los teléfonos móviles que están saliendo al mercado satisfacen esta restricción [6]. En particular, este trabajo se ha realizado con un iPhone 3G que incluye un acelerómetro que recoge las aceleraciones en los tres ejes del espacio en un rango de (-2.5g,2.5g).

La técnica biométrica de reconocimiento de firma en 3D puede considerarse como una combinación entre las técnicas habituales de comportamiento y físicas. La repetición de una firma en el aire no depende únicamente de características de comportamiento del usuario como la manera de sujetar el teléfono móvil, sino que además influyen una serie de características físicas que van a hacer que distintas personas puedan repetir un mismo gesto de manera distinta, como por ejemplo la longitud del brazo, la capacidad de girar la muñeca, el tamaño de la mano, etc. Esta técnica es similar al reconocimiento de usuarios por firma manuscrita [7], pero adaptada a un entorno de teléfonos móviles, con la ventaja de poder utilizarse los tres ejes del espacio, en vez de un único plano donde realizar la firma. De hecho, al no ofrecer un plano de referencia a posibles falsificadores, la imitación de la realización de una firma en el aire es más complicada.

De igual manera, esta técnica tiene aspectos comunes con las técnicas de reconocimiento de gestos, pero el enfoque es radicalmente distinto. Las técnicas de reconocimiento gestuales intentan reconocer un mismo gesto realizado por muchas personas distintas, que lo pueden hacer de manera diferente para después realizar una acción común a todos y en respuesta a ese gesto [8]. El enfoque en la técnica biométrica es diferenciar a la persona que realiza el gesto, así pues, si dos personas realizan el mismo gesto (o firma) en el aire, el sistema ha de ser capaz de identificar que los gestos, a pesar de su parecido, son distintos, pues corresponden a dos personas diferentes.

En esta propuesta, la extracción de características se realiza directamente en el propio móvil, sin ningún dispositivo adicional. Además, se pretende que todo el proceso de autenticación se realice también dentro del teléfono, para evitar los compromisos de seguridad en las conexiones con cualquier servidor externo. De esta manera, ejecutar todos los algoritmos involucrados en el proceso dentro del propio dispositivo móvil ofrece una gran cantidad de ventajas:

- El usuario no necesita gastarse más dinero en otros dispositivos, ya que únicamente necesita su propio teléfono móvil que ya tiene.
- Las posibilidades de ataque al sistema se reducen, ya que ninguna clave ni patrón sale fuera del dispositivo, ofreciendo una solución "Match on Card"[9].
- El sistema es resistente a ataques o caídas en las comunicaciones con un posible servidor externo que realice el proceso de autenticación.
- Esta configuración permite adoptar soluciones de criptobiometría, en el que la realización de una firma pueda generar, liberar o descifrar una clave asociada al usuario que se encuentra almacenada en el móvil, y que sólo él puede utilizar para realizar acciones que necesiten estar seguras de su identidad.

El proceso de autenticación de un usuario según esta técnica biométrica puede realizarse en un dispositivo móvil gracias al incremento de potencia de los microprocesadores de los mismos, que permiten ejecutar los algoritmos involucrados en una cantidad de tiempo razonables, logrando así alcanzar también el requisito de tiempo "real".

Por otro lado, la realización de todo el proceso de autenticación en el propio dispositivo móvil, permite aplicar de manera sencilla un modelo criptobiométrico de liberación de clave tras autenticación. De este modo, el teléfono móvil puede tener una clave criptográfica asociada a la identidad del usuario propietario del dispositivo. Para acceder a la clave y utilizarla en cualquier aplicación que necesite autenticación, el usuario ha de repetir su firma en el aire, asegurando así su identidad.

III. Método matemático de análisis

En este artículo se ha desarrollado un algoritmo basado en Programación Dinámica para analizar las diferencias entre señales para averiguar si una muestra se corresponde con la original o no. Este algoritmo propone buscar la alineación óptima de las dos señales que se quieren comparar, incluyendo una serie de ceros e interpolando las señales para tratar de maximizar una función de puntuación concreta [10]. Como consecuencia de este algoritmo, la longitud de las señales alineadas puede llegar a duplicarse. Una vez realizado el alineamiento óptimo de las señales, se calcula la distancia Euclídea entre cada par de señales, para cuantificar la diferencia en la realización de las firmas en el aire.

Este algoritmo incluye una función borrosa en la función de la puntuación que hay que maximizar representando la variabilidad con la que el propio usuario es capaz de repetir su firma en el aire. Esta ecuación de la puntuación del algoritmo se muestra en la Ecuación 1:

$$s_{i,j} = \max \begin{cases} s_{i,j-1} + h \\ s_{i-1,j-1} + \Delta \\ s_{i-i,j} + h \end{cases} \quad (1)$$

donde h es una constante conocida en la literatura como "gap penalty" [11] y cuyo valor se obtiene al maximizar el rendimiento global del sistema. El valor seleccionado es de $h = 0{,}4$, que coincide con un 12.5 % del rango de valores posibles de aceleraciones que el acelerómetro es capaz de extraer. Por otro lado, Δ es la función borrosa de decisión que representa una distribución Gaussiana:

$$\Delta = e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2)$$

donde μ y x son los valores de los puntos previos $(i-1, j-1)$ en base a los que se calcula la puntuación de los nuevos puntos (i, j). Esta función introduce una cierta borrosidad en la transición de un punto a otro que se refleja en la representación global de la variabilidad de las distintas realizaciones de un gesto por un mismo usuario. Finalmente, σ es una constate que representa hasta qué punto dos valores son similares. En este trabajo se ha elegido $\sigma = 0{,}250$ representando que dos puntos son iguales en un entorno del 10 % del rango de todos los puntos posibles de aceleración. En este tipo de técnica biométrica, un usuario no va a ser capaz de repetir al 100 % su firma en el aire; siempre existirá una pequeña variabilidad en la velocidad en la que el usuario realiza alguna parte de su firma, en la manera de realizar los giros y movimientos de su firmo o en el modo de agarrar el teléfono móvil. Gracias al método de análisis propuesto en esta Sección, estas pequeñas desviaciones pueden ser corregidas con facilidad sin compensar otras diferencias más notorias provenientes de las posibles falsificaciones de firmas por otros usuarios.

En la Sección V se presentarán las tasas de error obtenidas al utilizar este método para reconocer las firmas en el aire originales de distintos usuarios respecto a los intentos de falsificación de otros usuarios.

IV. Descripción de la base de datos utilizada

Los experimentos realizados en este artículo se han desarrollado a partir de una base de datos propietaria creada específicamente para este cometido. Han participado 34 usuarios que han realizado su firma en el aire y 3 falsificadores que han estudiado cada una de las firmas originales y han tratado de reproducirlas lo más fielmente posible. La base de datos final se ha obtenido en dos sesiones distintas:

En la primera sesión, cada usuario ha inventado un gesto identificativo y lo ha realizado en el aire con un teléfono móvil que integra un acelerómetro. Este gesto se corresponde con la firma en el aire que cada usuario ha seleccionado para utilizar en esta técnica biométrica propuesta. Para la primera sesión 34 usuarios diferentes (de edades entre 19 y 60 años, 15 mujeres y 19 hombres) han repetido 7 veces su firma en el aire, con intervalos de 10 segundos entre cada firma para reducir la dependencia entre muestras. Además, todas estas sesiones han sido grabadas en vídeo para que en la segunda sesión otros usuarios pudieran tratar de falsificar las firmas originales.

Para la toma de muestras y la extracción de características se ha desarrollado una aplicación para el iPhone 3G que registra las aceleraciones en los ejes X, Y y Z en la ejecución del gesto a una frecuencia de muestreo de 10ms, suficiente para obtener señales representativas del movimiento de la mano en el aire [12].

Al finalizar esta primera sesión, todos los voluntarios han respondido una encuesta para evaluar (1 muy bien - 5 muy mal) distintos aspectos de la técnica biométrica basada en reconocimiento de firmas en el aire propuestas. Los resultados se presentan en la Tabla I:

Pregunta	Media	Moda	Desviación estándar
Facilidad para inventar una firma en el aire	2.1	2	0.65
Facilidad para repetir una firma en el aire	1.9	2	0.45
Colectividad de la técnica	1.9	1	0.71
Aceptabilidad de la técnica	2.7	2	0.85

Cuadro I
RESPUESTAS DE LOS VOLUNTARIOS A DIFERENTES ASPECTOS PARA VALIDAR LA TÉCNICA BIOMÉTRICA BASADA EN RECONOCIMIENTO DE FIRMAS EN EL AIRE DESDE EL PUNTO DE VISTA DE LA EXPERIENCIA DEL USUARIO.

A partir de estas respuestas se puede inferir que para los usuarios ha sido bastante sencillo inventar y repetir una firma en el aire con un dispositivo móvil. Debido a que los datos biométricos se adquieren de manera no intrusiva, los usuarios han evaluado la colectividad [13] de la técnica como buena. Además, los usuarios se sienten seguros y cómodos cuando las características biométricas se extraen, por lo que la aceptabilidad de la técnica recibe también una buena nota.

Además, se les ha solicitado a los voluntarios que comparen la confianza que les ofrece esta técnica de reconocimiento de firmas en el aire respecto a otras técnicas biométricas como iris, cara, mano, huella dactilar y firma manuscrita. En media, los participantes han evaluado la confianza de la firma en el aire por encima de la firma manuscrita, ya que consideran que es más difícil de falsificar puesto que no hay un plano de referencia (papel) donde poder observar con facilidad los trazos de la firma. Además, la confianza que ofrece esta técnica a los usuarios es menor, pero próxima, a las técnicas de reconocimiento de cara y geometría de mano, y muy lejana a la confianza en las técnicas más robustas como iris y huella dactilar.

Por otro lado, a partir del estudio de los videos grabados en la sesión previa se ha realizado una segunda sesión para obtener muestras de intentos de falsificación de firmas originales. En esta sesión, tres usuarios han intentado falsificar cada una de las 34 firmas en el aire originales estudiando con detenimiento cada uno de los videos de la sesión anterior. Cada falsificador ha realizado 7 intentos de imitación de cada firma original.

Como resultado de ambas sesiones, se ha obtenido una base de datos de 238 muestras de gestos originales (34 usuarios $\times$ 7 repeticiones) y 714 falsificaciones (34 usuarios originales $\times$ 3 falsificadores $\times$ 7 intentos). A partir de esta base de datos, en la siguiente Sección se explicarán los experimentos que se han realizado y las tasas de error obtenidas para validar la fiabilidad de la técnica biométrica propuesta.

V. Resultados experimentales

En esta sección se presentan los experimentos que se han realizado analizando las señales obtenidas a partir de la base de datos de firmas en el aire creada mediante el método matemático propuesto en la Sección III.

Para medir la fiabilidad del sistema biométrico, se utilizarán las tasas de error típicas en biometría [14]: la tasa de error de

falsos positivos o aceptación de usuarios fraudulentos que han conseguido falsificar la firma de un usuario original (FAR, False Acceptance Rate), la tasa de error de falsos negativos o usuarios que a pesar de realizar su firma original el sistema ha rechazado (FRR, False Rejection Rate), y el punto donde se cortan las dos tasas anteriores, denominado EER (Equal Error Rate). Una tasa de EER suficientemente baja va a permitir utilizar esta técnica biométrica para la liberación de claves criptográficas que puedan estar en el teléfono móvil y sólo el usuario autenticado puede utilizar para asegurar su identidad en otras aplicaciones.

Para ello, se han seleccionado aleatoriamente tres muestras de cada gesto para conformar el patrón de la firma en el aire. Las otras cuatro muestras originales van a representar intentos originales de autenticación del usuario, que el sistema debe de aceptarlos. Todas las muestras de falsificaciones se consideran ataques al sistema que deben ser rechazados. Por tanto, las tasas de error EER se han calculado a partir de 136 muestras originales (34 usuarios $\times$ 4 muestras de acceso) y 714 muestras de impostores (34 usuarios $\times$ 3 falsificadores $\times$ 7 muestras).

Para poder evaluar esta técnica como potente, es imprescindible que las tasas de EER sean suficientemente bajas, pero también que el tiempo necesario para llevar a cabo la ejecución de los algoritmos involucrados en el proceso sea razonablemente corto. De acuerdo con esta idea, hay que remarcar que el tiempo de ejecución del algoritmo propuesto crece exponencialmente con la longitud de las señales, mientras que el incremento del número de ejecuciones del algoritmo con señales de longitud constante crece de manera lineal.

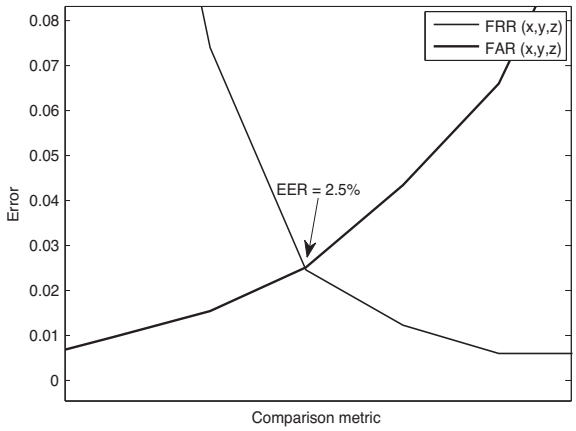
Cada firma en el aire almacena información sobre las aceleraciones en cada eje X, Y y Z producidas en la ejecución de la firma. Para fusionar estas tres señales se han probado tres estrategias de fusión multibiométricas diferentes, cada una en un nivel de la estructura biométrica [15]: en el nivel decisión ("Decision Level"), en el nivel de extracción de características ("Feature Extraction") o en el nivel de comparación ("Matching score"). En este artículo sólo se explicará la primera estrategia puesto que es la que ha producido mejores resultados.

Fusionar la información de las tres señales de aceleración de cada señal a nivel de decisión implica ejecutar en paralelo, pero de manera separada, el algoritmo de alineamiento para cada uno de los ejes y finalmente, calcular un único valor que cuantifique la diferencia entre las dos firmas a partir de los valores obtenidos en cada uno de los ejes. El valor de comparación para dos firmas en el aire A y B de longitud L se calcula según la Ecuación 3:

$$d_{A,B} = \frac{d^x_{A,B} + d^y_{A,B} + d^z_{A,B}}{3} \tag{3}$$

donde $d^x_{A,B}$, $d^y_{A,B}$ y $d^z_{A,B}$ son los valores obtenidos al alinear las señales en los ejes x, y y z de manera separada y calculando su distancia Euclídea según la Ecuación 4:

Figura 1. Equal Error Rate Resultante



$$d^e_{A,B} = \sum_{i=1}^{2L} (A'_{x,i} - B'_{x,i})^2 \tag{4}$$

donde A'_e y B'_e son las señales resultado de alinear las señales A y B en el eje e. Debido a que la longitud de las señales puede duplicarse en el algoritmo de alineamiento, el valor resultante $d^e_{A,B}$ para cada eje e se obtiene calculando las diferencias entre cada punto a lo largo de toda la longitud de las señales $2L$.

De acuerdo al escenario de fusión de información propuesto, el algoritmo de alineamiento se ejecuta tres veces, una para cada eje de manera separada. Los resultados del algoritmo se fusionan a nivel de decisión, calculando la media de los resultados de cada eje. Con estas condiciones, se ha obtenido un valor de EER (Equal Error Rate) de 2.5 % (Figura 1).

Sea T_E el tiempo de ejecución del algoritmo de alineamiento, que es la tarea que más tarda dentro del proceso de autenticación. Entonces, el tiempo consumido en este experimento es equivalente a tres veces la ejecución del algoritmo con dos señales de longitud L ($3T_E(L)$). Este tiempo se ha medido en un dispositivo móvil (iPhone 3G) resultando ser de 1.51 segundos en media. El cálculo de este tiempo se ha obtenido como la media de la ejecución del algoritmo con señales de 600 puntos por cada eje (una firma que dura seis segundos a una frecuencia de muestreo de 100 Hz) durante diez veces seguidas.

VI. Conclusiones y líneas futuras

En este artículo se ha propuesto una nueva técnica criptobiométrica para liberar una clave que se encuentra almacenada en un teléfono móvil mediante la realización de una firma en el aire con el mismo. Para ello, es necesario que el dispositivo incluya un acelerómetro para así poder extraer las aceleraciones en la ejecución de la firma. Con este sistema propuesto, todas las operaciones necesarias para la autenticación del usuario se procesan, en un tiempo razonable, dentro del propio teléfono móvil, sin necesidad de ningún dispositivo externo que encarecen el sistema e incluyen una comunicación que podría ser fuente de inseguridades.

Para estudiar la validez de esta técnica, se ha creado una base de datos de firmas biométricas en el aire. Para ello, 34 usuarios han inventado y repetido una firma en el aire con un teléfono móvil que integraba un acelerómetro. Además, en otra sesión, tres falsificadores han intentado imitar los gestos originales a partir de grabaciones de video. Los participantes en la creación de la base de datos han evaluado positivamente la facilidad, aceptabilidad, colectividad y confianza que les transmite la técnica biométrica propuesta.

A partir de la información de la técnica biométrica, se han estudiado diferentes escenarios de fusión de la información, obteniéndose las menores tasas de error cuando la fusión se llevaba a cabo a nivel de decisión. Para ello, se ha estudiado la tasa de Error de Falso Rechazo (FRR) a partir de las muestras originales y la tasa de Error de Falsa Aceptación (FAR) mediante las muestras de falsificaciones de los gestos originales. Como resultado de la intersección de las dos gráficas, se obtiene el valor final de Error (EER) de 2.5 %, que valida la seguridad de la técnica.

Una vez realizada la autenticación del usuario, la clave almacenada en el teléfono móvil y ligada a un único usuario puede ser liberada, permitiéndole al mismo la utilización de la misma para conectarse a aplicaciones que necesitan la seguridad de que el usuario es quien dice ser.

Además, se ha desarrollado una aplicación prototipo en un teléfono móvil que incluye un acelerómetro, desde el que directamente se ha medido que el proceso de autenticación propuesto en este artículo requiere 1.51 segundos. Teniendo en cuenta el tiempo en comunicarte con los servidores Web en dispositivos móviles o el tiempo en realizar una transacción por Internet desde un teléfono móvil, el tiempo obtenido en este estudio es perfectamente asumible por un usuario.

Como líneas de trabajo futuro en este tema aparece la posibilidad de aplicar otros esquemas criptobiométricas más complejos a la técnica biométrica de reconocimiento mediante firma en el aire. Estudiando distintas características intrínsecas a las señales de aceleración en cada eje en la repetición de una firma, podría generarse automáticamente una clave criptográfica, mejorando de en gran medida el modelo puesto que ya no sería necesario almacenar una clave ligada a un usuario en su teléfono, sino que el usuario mismo sería capaz de generarla cuando la necesitara realizando su firma.

REFERENCIAS

[1] D. ho Cho, K. R. Park, D. W. Rhee, Y. Kim, and J. Yang, "Pupil and iris localization for iris recognition in mobile phones," *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, International Conference on & Self-Assembling Wireless Networks, International Workshop on*, vol. 0, pp. 197–201, 2006.

[2] Q. Tao and R. Veldhuis, "Biometric authentication for a mobile personal device," *Mobile and Ubiquitous Systems, Annual International Conference on*, vol. 0, pp. 1–3, 2006.

[3] H. A. Shabeer and P. Suganthi, "Mobile phones security using biometrics," *Computational Intelligence and Multimedia Applications, International Conference on*, vol. 4, pp. 270–274, 2007.

[4] H. Manabe, Y. Yamakawa, T. Sasamoto, and R. Sasaki, "Security evaluation of biometrics authentications for cellular phones," *Intelligent Information Hiding and Multimedia Signal Processing, International Conference on*, vol. 0, pp. 34–39, 2009.

[5] F. Hao, R. Anderson, and J. Daugman, "Combining crypto with biometrics effectively," *IEEE Trans. Comput.*, vol. 55, no. 9, pp. 1081–1088, 2006.

[6] J. H. Steve Dowling, Nancy Paxton, "Apple reports first quarter results," Apple Inc., Tech. Rep., 2009. [Online]. Available: http://www.apple.com/pr/library/2009/01/21results.html

[7] A. J. Friederike, A. K. Jain, F. D. Griess, S. D. Connell, E. Lansing, and M. J, "On-line signature verification," *Pattern Recognition*, vol. 35, 2002.

[8] J. Daugman, "Face and gesture recognition: Overview," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 675–676, 1997.

[9] J. Nilsson and M. Harris, "Match-on-card for java cards," Precise Biometrics, Tech. Rep., 2004.

[10] A. de Santos Sierra, C. Avila, and V. Vera, "A fuzzy dna-based algorithm for identification and authentication in an iris detection system," in *Security Technology, 2008. ICCST 2008. 42nd Annual IEEE International Carnahan Conference on*, Oct. 2008, pp. 226–232.

[11] W. Miller, N. C. Jones, and P. A. Pevzner, "An introduction to bioinformatics algorithms." *Journal of the American Statistical Association*, vol. 101, pp. 855–855, June 2006. [Online]. Available: http://ideas.repec.org/a/bes/jnlasa/v101y2006p855-855.html

[12] C. Verplaetse, "Inertial proprioceptive devices: self-motion-sensing toys and tools," *IBM Syst. J.*, vol. 35, no. 3-4, pp. 639–650, 1996.

[13] A. Jain, L. Hong, and S. Pankanti, "Biometric identification," *Commun. ACM*, vol. 43, no. 2, pp. 90–98, 2000.

[14] A. K. Jain, P. Flynn, and A. A. Ross, *Handbook of Biometrics*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[15] A. Ross and A. Jain, "Information fusion in biometrics," *Pattern Recognition Letters*, vol. 24, no. 13, pp. 2115–2125, September 2003.

Una metodología para la protección mutua automática de sistemas multiagentes

Pablo Antón, Antonio Muñoz, Antonio Maña
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga
Email: {panton,amunoz,amg}@lcc.uma.es

***Resumen*—El paradigma de los agentes móviles representa una de las tecnologías más prometedoras para los escenarios de Ambientes Inteligentes donde un gran numero de dispositivos interactúan. Desafortunadamente, la carencia en los mecanismos de seguridad, tanto en la aplicación de estos mecanismos como en su usabilidad, está dificultando la aplicación de este paradigma a las aplicaciones del mundo real. En el artículo que presentamos a continuación mostramos una solución software para la protección de sistemas multiagentes. Nuestra propuesta está enfocada en un modelo de agentes cooperativos y se basa en la computación protegida. Por último, uno de los atractivos más importantes del trabajo realizado es la facilidad de uso para aquellos desarrolladores de sistemas basados en agentes que no sean expertos en seguridad.**

I. Introducción

En el área de los sistemas de información, la seguridad es uno de los aspectos en los que se requiere mayor esfuerzo. Con el auge que tanto en la década de los 90 como en los últimos años han experimentado los sistemas distribuidos, se ha producido asimismo un aumento de los ataques informáticos y, por tanto, de los sistemas de protección.

El presente artículo se centra en los sistemas de agentes móviles inteligentes y en los esquemas de seguridad mutua estática [9].

Los primeros trabajos sobre agentes software surgen a mediado de los años 70 de la mano de Carl Hewitt [4]. Hewitt creó un modelo de agente (llamado actor) al que define como un objeto autónomo que interacciona y se ejecuta concurrentemente y que posee un estado interno y capacidad de comunicación.

En la actualidad existen multitud de variantes de agentes software dependiendo de sus características, habilidades o propiedades. Nuestro trabajo va enfocado a un tipo de agentes software conocido como agentes móviles y a su seguridad.

Los agentes móviles son implementaciones de programas remotos, es decir, programas que se desarrollan en una máquina y se distribuyen en otras máquinas para ejecutarse posteriormente [10]. La capacidad de migración provoca diferentes riesgos de seguridad y hace necesario que se controlen los siguientes aspectos:

- protección de la máquina contra los agentes
- protección de los agentes contra la máquina
- protección de la red.

Existen diversas propuestas de protección para cada uno de los puntos descritos anteriormente, nosotros sólo hemos tratado las relacionadas con la protección de los agentes contra la máquina. En este caso, se ha de suponer que no se puede confiar en las máquinas(agencias) a las que los agentes móviles pueden migrar, lo cual provoca riesgos de seguridad en todo el sistema multiagente.

Una de las estrategias para solventar estos problemas y añadir un mayor grado de seguridad al sistema multagente se basa en las ideas de la computación protegida [8]. Dicha idea, propone dividir el código en dos o más partes, algunas de estas partes se ejecutarán en un procesador seguro y confiable, mientras que las otras partes podrán ejecutarse en cualquier procesador.

Al aplicar esta estrategia de seguridad a los sistemas multiagentes se logra un modelo en el que cada agente colabora con uno o más agentes remotos, los cuales a su vez se ejecutan en diferentes agencias confiables o no. Para que un ataque sea efectivo necesita la cooperación de todas las agencias. Podemos distinguir dos esquemas diferentes dentro de la protección mutua:

- protección mutua estática, que es la que utilizamos en este trabajo y
- protección mutua dinámica.

Gracias a los trabajos de investigación desarrollado por miembros de la Universidad de Málaga se han creado las librerías necesarias que implementan las ideas de protección mutua. Siguiendo esta linea de investigación nos planteamos el objetivo de crear una herramienta que, utilizando dichas librerías, sea capaz de generar automáticamente un sistema multiagente con seguridad mutua estática a partir de un sistema multiagente sin seguridad. Esto nos permitiría comprobar fácilmente el funcionamiento y la eficacia del sistema dependiendo de los parámetros que determinan el nivel de seguridad y escogiendo en cada caso el más apropiado para el sistema multiagente original.

El presente documento está organizado de la siguiente forma: En la sección 2 comentamos los trabajos relacionados, introduciéndonos en los sistemas multiagentes móviles, plataforma Jade y esquemas de seguridad. La sección 3 se centra en el problema principal de este articulo, generar automáticamente un sistema multiagente con seguridad mutua estática. Describimos en la sección 4 la arquitectura y características del software desarrollado y por último introducimos una sección con las conclusiones del trabajo.

II. Trabajos relacionados

Anteriormente mencionamos la existencia de diferentes propuestas para la ejecución segura de software. Vamos a centrarnos en las técnicas que establecen una protección bidireccional. Entre ellas, la llamada Computación Confiable, que se basa en crear un sistema en el que la seguridad de todos los elementos se compruebe a partir de un componente confiable, frecuentemente construido en hardware [11]. A partir de esta idea se construyó un modelo alternativo denominado Computación Protegida [8], [7]. Dicha propuesta se basa en la idea de dividir el código en dos o más partes. Algunas de ellas se ejecutarán en un procesador seguro, mientras que otras se ejecutarán en un procesador cualquiera. De forma que no sea posible ejecutar la aplicación sin la colaboración del procesador confiable.

La división del código en partes mutuamente dependiente debe conseguir que:

- La parte pública no sea de utilidad para obtener información de la protegida
- Un trazado de la comunicación entre ambas partes no pueda usarse para obtener información de la parte protegida.

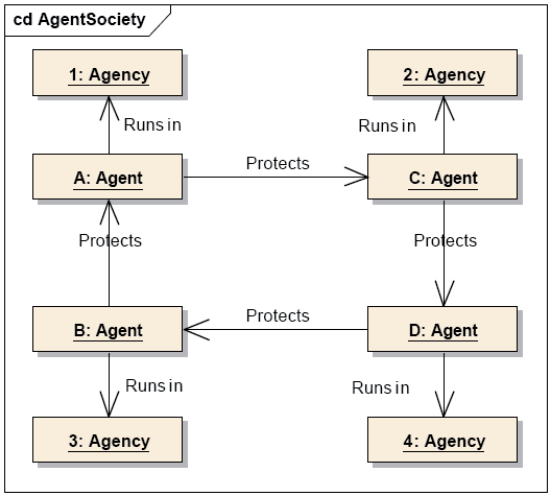


Figura 1. Una sociedad de agentes con protección mutua.

Este esquema de seguridad puede ser usado en distintos tipos de escenarios. Por ejemplo, un sistema multiagente con agentes móviles que se trasladen a diferentes agencias (no confiables) para llevar a cabo algunas tareas colaborativas. Debido al desconocimiento de las agencias no se puede garantizar la correcta ejecución del agente y su integridad, por ello el principal objetivo en este escenario será proteger a los propios agentes frente a posibles agencias maliciosas. En la Figura 1 podemos observar como se relacionan entre sí ejecutándose en diferentes agencias. Los agentes se protegerán unos a otros y por turnos. En esta configuración un ataque requerirá la cooperación de todas las agencias. Para este ejemplo específico es posible que las partes protegidas de un agente estén incluidas directamente en otros agentes, lo que dificulta la posibilidad de que se produzca la transmisión de las secciones de código protegido por la red.

Vamos a destacar dos esquemas diferentes a la hora de aplicar esta estrategia de seguridad. En la primera, a la que llamaremos protección mutua estática, la colaboración entre los agentes está predefinida. Esto significa que cada agente posee el código privado de uno o más agentes con los que colabora. Por otro lado tenemos la protección mutua dinámica que hace posible que los agentes del sistema funcionen como coprocesadores seguros para los demás agentes. En este caso, la interacción entre los agentes no está predefinida.

II-A. Protección mutua estática

Para el desarrollo de nuestra herramienta nos hemos centrado en la protección mutua estática. Para este tipo de estrategia las partes protegidas de un agente están directamente incluidas en él o en los agentes protectores. En la Figura 2 podemos observar una abstracción de la un agente con protección mutua estática.

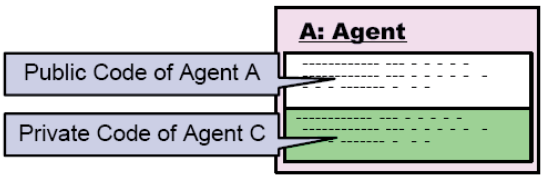


Figura 2. Estructura de un agente con protección mutua estática.

La gran ventaja de esta estrategia es que incrementa el rendimiento del sistema gracias a que no se producen trasmisiones de código protegido a través de la red. Por el contrario, su gran limitación exige que el sistema sea configurado estáticamente y los agentes deban ser protegidos antes de su ejecución.

Estos serían los trabajos teóricos en los que se basa nuestra aplicación, pero falta una base práctica y herramientas que implementen todos estos conceptos. La plataforma para trabajar con agentes la conseguimos gracias a JADE [12] y con la librería SecureAgent le añadimos el esquema de seguridad mutua estática.

II-B. La librería SecureAgent

La implementación de seguridad mutua estática que utilizamos ha sido desarrollada por Manuel Jiménez en su proyecto fin de carrera bajo la tutela de Antonio Maña y Antonio Muñoz. El lector puede encontrar más detalles de la librería en [5].

Para construir un sistema multiagente con las propiedades y funcionalidades de seguridad mutua estática, basta con heredar de la clase SecureAgent. Los agentes que heredan de esta clase poseen comportamientos que se encargan de realizar las tareas de protección. Al utilizar la librería no es necesario conocer la función de cada uno de los comportamientos.

Además de los comportamientos, es necesario mencionar la interfaz PrivateCode. Es una interfaz en la que sólo se define un método. De esta forma, el fragmento de código

privado de cada agente irá en el método execute() de la clase específica que implemente la interfaz PrivateCode. El agente deberá realizar una llamada al método execute() con los parámetros correspondientes para comenzar la ejecución del código privado.

En la Figura 3 podemos ver como sería el intercambio de mensajes entre dos agentes para realizar una ejecución segura.

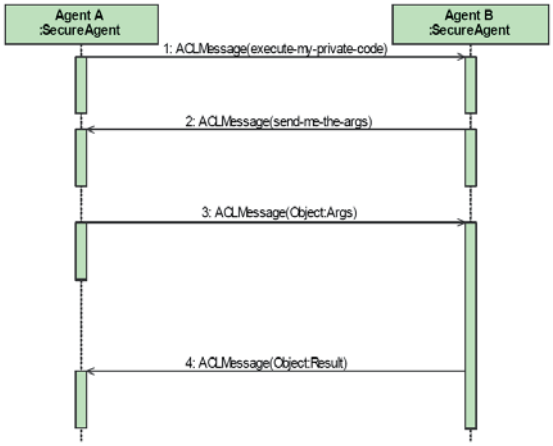


Figura 3. Intercambio de mensajes para una ejecución segura.

III. Nuestra solución

Antes de explicar la herramienta que hemos desarrollado vamos a hacer un resumen de la situación. Por un lado tenemos un sistema de agentes móviles inteligentes y la plataforma de agentes Jade, y por otro lado contamos con la estrategia de seguridad mutua estática y la librería SecureAgent con la que conseguimos un sistema de agentes seguros.

Por lo tanto, a partir del sistema de agentes móviles y utilizando las librerías de seguridad mutua estática el programador debe ser capaz de crear un sistema multiagente equivalente al inicial pero con seguridad incorporada. Este proceso implica conocer las librerías, reprogramar las clases y seleccionar los datos y código que queremos proteger.

Quizás, realizar estos pasos 1 o 2 veces no sería muy tedioso, pero es poco práctico. Imaginemos que después de trabajar unos horas realizando esta tarea carguemos los nuevos agentes para comprobar su funcionamiento y comprobemos que su eficiencia es baja o que el nivel de seguridad no es el adecuado.

Lo interesante es poder cambiar rápidamente la configuración de seguridad, ver los resultados y decidir si son los más adecuados para el sistema en cuestión. Por todo ello, el objetivo de nuestra herramienta es agilizar y automatizar todo este proceso.

Con el escenario inicial y el objetivo principal perfectamente definido comenzamos a estructurar los requisitos del sistema y a seleccionar las herramientas necesarias. El esquema general del sistema lo podemos ver en la Figura 4.

Como podemos observar el punto de partida o entradas de nuestra aplicación son los agentes sin seguridad que vendrán

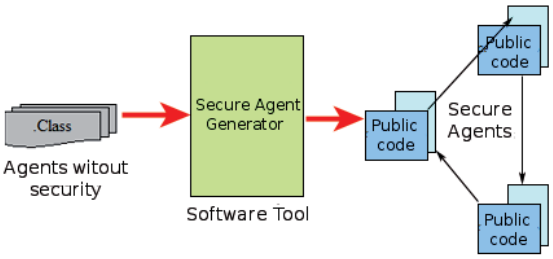


Figura 4. Arquitectura del generador de agentes seguros.

definidos por unas clases java (archivos .class). Las salidas del programa son estos mismos agentes pero con seguridad incorporada y de igual forma definidos por unas clases java.

Por lo tanto el trabajo que tenemos que realizar es leer, analizar, modificar y crear archivos .class. Para ello necesitamos una herramienta que nos permita trabajar cómodamente con este tipo de ficheros. Existen varias herramientas que tienen estas características como BCel [3], Javassist [1] o ASM [2], nosotros optamos por la primera ya que cuenta con bastante documentación, está desarrollada por Apache Software Foundation, lo cual nos da cierta confianza y es multi-plataforma al estar escrita en java.

III-A. Byte Code Engineering Library (BCEL)

Un fichero .class tiene una estructura interna bastante compleja y difícil de manejar como se muestra en la Figura 5. Sus numerosas referencias y el bajo nivel del código que contiene hacen de su análisis y creación una labor muy tediosa.

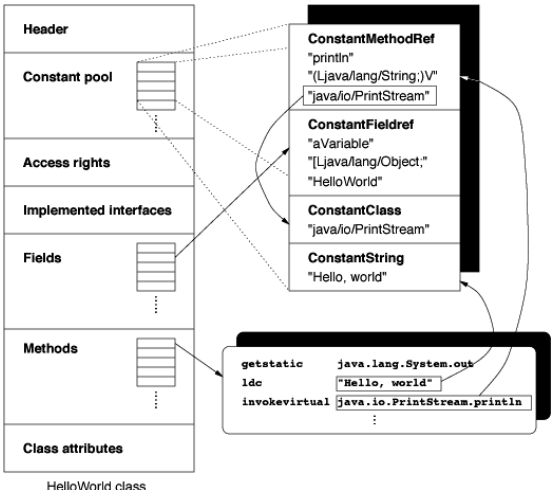


Figura 5. Formato de un archivo .class.

Para trabajar con estos ficheros usamos Bcel, un proyecto desarrollado por Apache Software Foundation y que nos ofrece una amplia API que podemos dividir en tres partes:

- Un paquete que contiene clases que describen las propiedades "estáticas" de los ficheros de clase*. Las clases deben ser utilizadas para leer y escribir ficheros de clase desde o hacia un archivo. Esto es especialmente

útil para analizar clases Java sin tener el código fuente a mano. La clase principal de este paquete es JavaClass.
- Un paquete para generar o modificar dinámicamente ficheros de clase. Puede ser utilizado para añadir o analizar código de los ficheros de clase, etc.
- Varios códigos de ejemplo y utilidades, como un visor de ficheros de clase, una herramienta para convertir fichero de clase a HTML y al lenguaje ensamblador Jasmin.

El componente estático consta de una serie de clases que modelan la estructura interna de un fichero .class. La clase principal, denominada JavaClass, se construye a partir de un fichero .class y ofrece multitud de funciones para navegar por sus diferentes componentes, campos, métodos, variables locales, clases internas, etc.

Además proporciona diversos patrones para el control y el análisis de los elementos como pueden ser el patrón visitante o el observador. En nuestro caso, y para el análisis de las instrucciones, el patrón visitante ha sido de gran utilidad. La jerarquía de instrucciones bytecode se ajusta muy bien a este patrón y te permite organizar cómodamente qué hacer con cada una de ellas. Por ejemplo, en una instrucción LOAD/STORE será necesario conocer qué campo se quiere leer o, en una INVOKE el método al que se invoca.

Este componente estático no nos permite modificar ni crear nuevos ficheros .class que es primordial para nuestro proyecto. Para ello hacemos uso del componente dinámico. Su clase principal es ClassGen que nos otorga la posibilidad de crear una clase vacía e ir añadiéndole dinámicamente todos los componentes que deseemos, para finalmente guardarla como un archivo .class.

Por lo tanto ya tenemos la herramienta que nos va a permitir por una lado analizar los archivos de entrada que representan a los agentes sin protección y por otro generar las nuevas clases que representaran a los agentes seguros utilizando la librería SecureAgent.

IV. Arquitectura

En esta sección vamos a mostrar con mayor detalles las características y el funcionamiento de la herramienta que hemos desarrollado. Como comentamos anteriormente el objetivo principal del software creado es generar automáticamente un sistema de agentes con seguridad mutua estática a partir de un conjunto de agentes sin seguridad.

Los ficheros de entrada que corresponde a los agentes sin seguridad deben cumplir ciertas restricciones (precondiciones) entre las que podemos comentar:

- deben ser archivos precompilados java .class
- deben representar a una clase que extienda de jade.core.Agent.
- no pueden contener clases internas anónimas.

De igual forma existen una serie de condiciones de salida que debemos tener muy en cuenta:

- cada uno de los nuevos agentes seguros tendrá asignado un agente protector, este agente protector no puede ser modificado en tiempo de ejecución.
- el comportamiento del nuevo sistema de agentes con la seguridad incorporada será igual al del sistema original.

La arquitectura general del sistema sigue el patrón Modelo-Vista-Controlador. Pero, ya que la vista es una simple interfaz gráfica de usuario para facilitar el uso de la herramienta, sólo vamos a mostrarla en la Figura 6 y nos centraremos en el modelo de datos y en sus fases de funcionamiento.

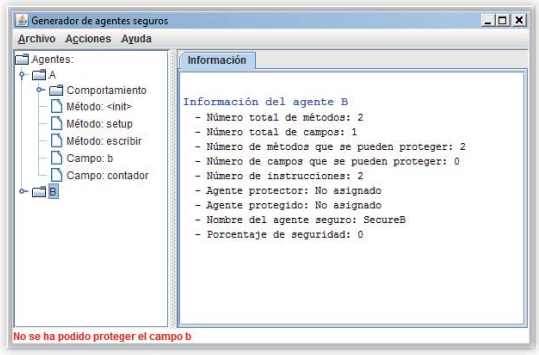


Figura 6. Interfaz gráfica de usuario.

Podemos dividir el funcionamiento de la herramienta en tres fases bien diferenciadas: carga de agentes originales, configuración de seguridad y creación de agentes con seguridad. Para cada una de estas fases se han implementado una serie de clases que se encargan de su correcto funcionamiento.

Creemos que es mejor ilustrar todo el proceso con un ejemplo práctico, en el podremos observar con facilidad cada una de las etapas.

La clase de la Figura 7 contiene el código de un agente JADE sin seguridad, como podemos observar hereda de la clase Agent y su código de ejecución está contenido en el método setup().

```
public class Ejemplo extends Agent{

   // Campos
   ...
   protected void setup() {
      ...
      // Instrucción que no se protegerá
      System.out.println("No protegida");

      // Instrucción que si se protegerá
      System.out.println("Protegida");
      ...
   }
}
```

Figura 7. Código de ejemplo.

Puesto que es necesario que cada agente tenga un agente protector, el mínimo número de agentes para que el sistema tenga sentido es dos. Supongamos entonces que en este ejemplo también contamos con un agente vacío (sin instrucciones ni datos), que hará la labor de agente protector.

A continuación vamos a explicar cada una de las fases por separado y nos centráremos en el ejemplo dado.

IV-A. Fase I: Carga

Esta es la primera fase en el proceso de generación de los agentes seguros. Su cometido es el de cargar los ficheros de clase y analizar todo su contenido. Para ello debemos seleccionar los agentes sin seguridad e identificar y analizar todos sus elementos, es decir, métodos, campos, instrucciones, clases internas, etc.

En esta etapa de análisis de los ficheros .class utilizamos el componente estático de las librerías de BCEL. Como ya hemos comentado con anterioridad, esta parte de la API nos permite cargar un archivo .class y generar automáticamente la estructura de un fichero de clase [6]. Por cada uno de los elementos que conformar el fichero de clase (métodos, campos, clases internas, instrucciones, etc) se crea un objeto que lo modela y nos permite trabajar con él.

Todos los elementos generados por este componente de BCEL son de solo lectura. Pero necesitamos guardar información de cada uno de estos elementos. Para ello hemos creado clases que heredan directamente de las clases que nos ofrece BCEL. En estas nuevas clases introducimos toda la información que utilizaremos en los procesos posteriores.

Dentro de este proceso de análisis vamos a destacar y aclarar el caso de las instrucciones, ya que este es más complejo que los demás elementos. Como podemos observar en la Figura 5, dentro de los ficheros de clases hay una sección dedicada a los métodos de la clase. Entre otros elementos, dentro de estos métodos nos encontramos instrucciones bytecode.

Estas instrucciones suelen carecer de sentido si se ejecutan por separado, es decir, que dependen de la antecesora y/o la sucesora. En la Figura 8 podemos observar la correspondencia entre una instrucción en java y el conjunto de instrucciones en bytecode. Por ello tomamos la decisión de agruparlas en conjuntos que correspondiesen a una instrucción java terminada en ;.

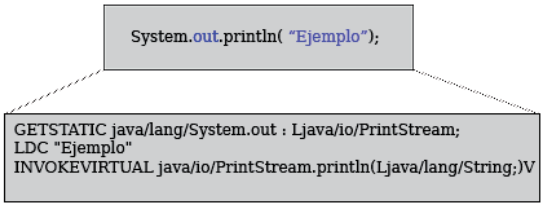


Figura 8. Correspondencia.

Una vez que estén cargados todos los agentes en el sistema pasamos a la fase de configuración, donde vamos a indicar el grado de seguridad y los enlaces de protección.

IV-B. Fase II: Configuración

La segunda fase del proceso es la más simple de las tres y la que ha resultado más fácil de implementar. Se trata de especificar los parámetros de seguridad con los que se crearán los nuevos agentes seguros.

Existen dos elementos susceptibles a ser protegidos en un agente JADE:

- instrucciones
- datos (Campos de la clase)

La información necesaria para determinar el grado de seguridad será el porcentaje de instrucciones y los datos que queremos proteger. Toda esta información se modela con una clase llamada SecureAgentConfiguration. Se han implementado dos maneras de indicar los parámetros de seguridad.

Con el primer método es necesario especificar por cada uno de los agentes cargados (en la primera fase) en el sistema los parámetros de seguridad. Esto implica seleccionar uno a uno los agentes e introducir el porcentaje de instrucciones y seleccionar los campos que queremos proteger.

Este método puede ser poco fluido dependiendo del número de agentes cargados y el número de pruebas que deseemos realizar. Por ello, hemos implementado una opción que permite aplicar una plantilla de seguridad a todos los agentes cargados a través de un archivo XML. Existen tres plantillas básicas y la posibilidad de cargar la plantilla personal desde un archivo. En la Figura 9 podemos ver un ejemplo del formato de un fichero de configuración. En él podemos observar el porcentaje de instrucciones y de datos, 50 en este caso, que deseamos proteger.

```
<?xml version="1.0" encoding="windows-1250"?>

<configuracion>
   <instrucciones> 50 </instrucciones>
   <datos> 50 </datos>
</configuracion>
```

Figura 9. Formato del fichero XML.

Es importante tener en cuenta que existen casos en los que estos porcentajes no se pueden cumplir con exactitud. En ocasiones no es viable proteger una instrucción o un dato, por ejemplo, los datos estáticos o las instrucciones de salto.

Antes de pasar a la siguiente etapa y para concluir con la configuración debemos seleccionar cómo van a ser los enlaces de protección, es decir, indicar cómo van a protegerse los agentes entre ellos. No es necesario realizar esta acción manualmente mano ya que existe una opción para automatizarla en la aplicación gráfica. En nuestro caso, al existir únicamente dos agentes, se protegerán el uno al otro.

IV-C. Fase III: Creación de los agentes seguros

Por último, tenemos la fase de creación de los agentes seguros. Gracias a los pasos anteriores en esta fase ya contamos con toda la información necesaria para la creación.

Por cada uno de los agentes originales debemos crear al menos dos nuevas clases, una corresponderá al nuevo agente seguro que contendrá su código (datos e instrucciones) público y la otra el código privado. Además de estas clases se crearán tantas como clases internas contenga el agente original.

Para crear estas nuevas clases precompiladas hemos utilizado el componente dinámico de BCEL. Con esta parte de la

API de BCEL creamos el esqueleto de los ficheros de clase y dependiendo de los parámetros de seguridad que se hayan establecido en la fase anterior introducimos él código original en una clase u otra.

En la Figura 10 podemos observar el contenido de la parte publica del nuevo agente seguro basado en el ejemplo de la Figura 7. Se ha modificado la clase de la siguiente manera:

```
public class SecEj extends SecureAgent{

   protected void setup() {

      // Inicialización

      // Instrucción no protegida
      System.out.println("No protegida");

      // Llamada al código remoto
      ACLMessage msg = new ACLMessage(...);
      msg.setContent("execute-my-...");
      myArgs = argument;
      msg.addReceiver(this.protectedBy);
      send(msg);
   }
}
```

Figura 10. Agente seguro.

- La nueva clase hereda de SecureAgent.
- Se añade una sección de inicialización en el método setup donde se indica el agente protector y el protegido.
- Se añade el código no protegido.
- Se añaden solicitudes de ejecución de código remoto. Por cada sección de código que se haya protegido será necesario añadir las instrucciones para configurar los argumentos, realizar la llamada, esperar y recoger los resultados si los hubiera.

Para el código protegido se crea otra clase que implementa la interfaz PrivateCode. En esta nueva clase debemos introducir:

- Los campos protegidos (ninguno en nuestro caso).
- El método execute() que contendrá el código protegido dividido por secciones. La información para saber qué sección se debe ejecutar estará contenida en los argumentos del método. En nuestro caso y ya que el código a proteger cuenta con una única sección, esta estará directamente en el método execute() como muestra la Figura 11.

```
public class Priv implements PrivateCode{

   // Campos protegidos
   ...
   public Object exucute(Object o){

      // Instrucción que si se protegerá
      System.out.println("Protegida");
      return null;
   }
}
```

Figura 11. Codigo privado/protegido.

V. Conclusiones

A lo largo de este artículo hemos presentado una metodología que, mediante el uso de ciertas herramientas de apoyo, nos permiten la protección automática de sistemas multiagentes. Todo lo presentado y a pesar de que todas las herramientas de apoyo son totalmente funcionales, es un trabajo académico. Por ello no se contemplan algunos características de los sistemas multiagentes como puede ser el dinamismo de los agentes. El siguiente paso es lógico y consistiría en el estudio de una metodología paralela aplicable a sistemas reales dinámicos. El estudio de esta metodología y su despliegue está en etapas avanzadas y sus resultados serán presentados en próximos trabajos.

Referencias

[1] S. Chiba. *Javassist (Java Programming Assistant)*. Sun Microsystems, Inc., 2009.
[2] OW2 Consortium. *ASM*.
[3] Apache Software Foundation. *BCEL (Byte Code Engineering Library)*, 2006.
[4] C. Hewitt and H. Baker. Actors and continuous functionals, 1977.
[5] Manuel Jimenez. Computación protegida aplicada a la seguridad de sistemas multiagentes, 2007.
[6] T. Lindholm and F. Yellin. *The JavaTM Virtual Machine Specification*. Sun Microsystem, 1999.
[7] A. Maña, Javier Lopez, Juan J. Ortega, Ernesto Pimentel, and Jose M. Troya. A framework for secure execution of software. *International Journal of Information Security*, 3(2):99–112, octubre 2004.
[8] A. Maña, A. Muñoz, and D. Serrano. Computación Confiable frente a Computación Protegida. In *IX Reunión Española sobre Criptología y Seguridad de la Información (RECSI'06)*, Barcelona, 2006. UOC Press.
[9] A. Maña, A. Muñoz, and D. Serrano. Towards secure agent computing for ubiquitous computing and ambient intelligence. In *Lecture Notes in Computer Science*, volume 4611, page 1201–1212. Springer, 2007.
[10] H.S. Nwana. Software agents: An overview, 1996.
[11] Siani Pearson. Trusted computing platforms, the next security solution. *HP Labs*, 2002.
[12] Telecom Italia SpA. *JADE (Java Agent DEvelopment Framework)*, 2009.

Integración de RadSec y DAMe sobre eduroam

Francisco J. Moreno[1], Manuel Gil Pérez[1], Gabriel López[1],
Antonio F. Gómez Skarmeta[1], Sascha Neinert[2]
[1]Departamento de Ingeniería de la Información y las Comunicaciones, Universidad de Murcia
Email: {fran.moreno, mgilperez, gabilm, skarmeta}@um.es
[2]Computing Centre, Universidad de Stuttgart
Email: sascha.neinert@rus.uni-stuttgart.de

***Abstract*—Actualmente la red eduroam está formada por una jerarquía de servidores RADIUS que permite la movilidad de los usuarios a través de las organizaciones pertenecientes a esta federación. Este trabajo describe cómo esta jerarquía puede ser mejorada mediante el uso de propuestas como RadSec y DAMe, para ofrecer mayor seguridad en la conexión y en el acceso a servicios. Este trabajo muestra una integración de ambas propuestas sobre eduroam y un análisis de rendimiento para comprobar su viabilidad.**

***Index Terms*—RadSec, DAMe, eduroam, federación.**

I. Introducción

Bajo el marco de TERENA se ha desplegado una de las mayores redes para roaming a nivel mundial, *eduroam* [1]. Esta red, orientada a instituciones relacionadas con la investigación y la educación, permite la movilidad de usuarios a través de más de 40 países a lo largo de 3 continentes, incluyendo China, Australia, Canadá y próximamente EEUU.

Para conseguir esta cobertura, eduroam hace uso de una jerarquía distribuida de servidores RADIUS a lo largo de cada zona geográfica, país e institución. De este modo, por ejemplo, cuando un usuario de la Universidad de Stuttgart realiza una solicitud de acceso a la red eduroam, durante su estancia en la Universidad de Calgary, en realidad lo que ocurre es que está solicitando acceso al servidor RADIUS de la universidad visitada. Este servidor redirige la solicitud al nivel superior de la jerarquía, y así sucesivamente, hasta que se alcanza el servidor RADIUS de la universidad local del usuario. En el caso de conexiones intercontinentales pueden requerirse hasta 6 servidores intermedios para una autenticación del usuario. Otro de los problemas conocidos de este tipo de jerarquías es que la comunicación entre servidores se basa en conexiones UDP, protegidas a través de secretos compartidos.

Para ofrecer mayor seguridad en las comunicaciones y mayor control sobre el proceso de autenticación del usuario se han definido varias propuestas. Por un lado, RadSec [2] propone la extensión de RADIUS para el soporte de conexiones TCP/TLS; además, ofrece mecanismos de descubrimiento de la organización local del usuario, lo que permite evitar recorrer toda la jerarquía. Por otro lado, el proyecto DAMe [3] define cómo puede extenderse la jerarquía de eduroam para permitir servicios de *Single Sign On* (SSO) y control de acceso avanzado basado en atributos del usuario.

Este trabajo presenta cómo ambas soluciones pueden cooperar para ofrecer de un modo homogéneo todos los servicios de seguridad ofrecidos por ambas propuestas. Además, se presenta un análisis de rendimiento que permite comparar cómo el uso de canales TLS y mecanismos de descubrimiento pueden afectar al rendimiento de eduroam.

Para describir esta integración, este trabajo está estructurado del siguiente modo. La Sección II describe la propuesta de DAMe, mientras que la Sección III hace una breve introducción de RadSec. La Sección IV describe la integración de ambas tecnologías, indicando los componentes y mecanismos de descubrimiento. La Sección V presenta el análisis de rendimiento y la Sección VI describe brevemente el trabajo relacionado. Finalmente, la Sección VII presenta algunas conclusiones y vías futuras.

II. DAMe

La arquitectura general de DAMe se puede ver en la Figura 1, donde se muestra que eduroam es la iniciativa central de este proyecto. La principal idea que se pretende conseguir con DAMe es desplegar sobre eduroam el intercambio de credenciales de autorización y ofrecer mecanismos de SSO, ya desde el acceso a la red. Con DAMe, cualquier usuario será autenticado inicialmente usando eduroam, obteniendo un token SSO (eduToken) que será utilizado posteriormente para acceder a otros recursos de la federación.

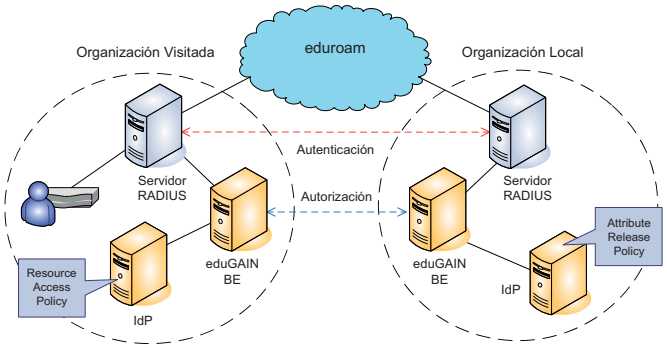


Fig. 1. Arquitectura general de DAMe

Respecto al proceso de autorización, esta arquitectura hace uso de *eduGAIN* [4] para ofrecer gestión de credenciales a nivel de federación sobre eduroam. De este modo, permite utilizar el sistema como *back-end* de control de acceso a la federación, mientras que cada organización de forma local

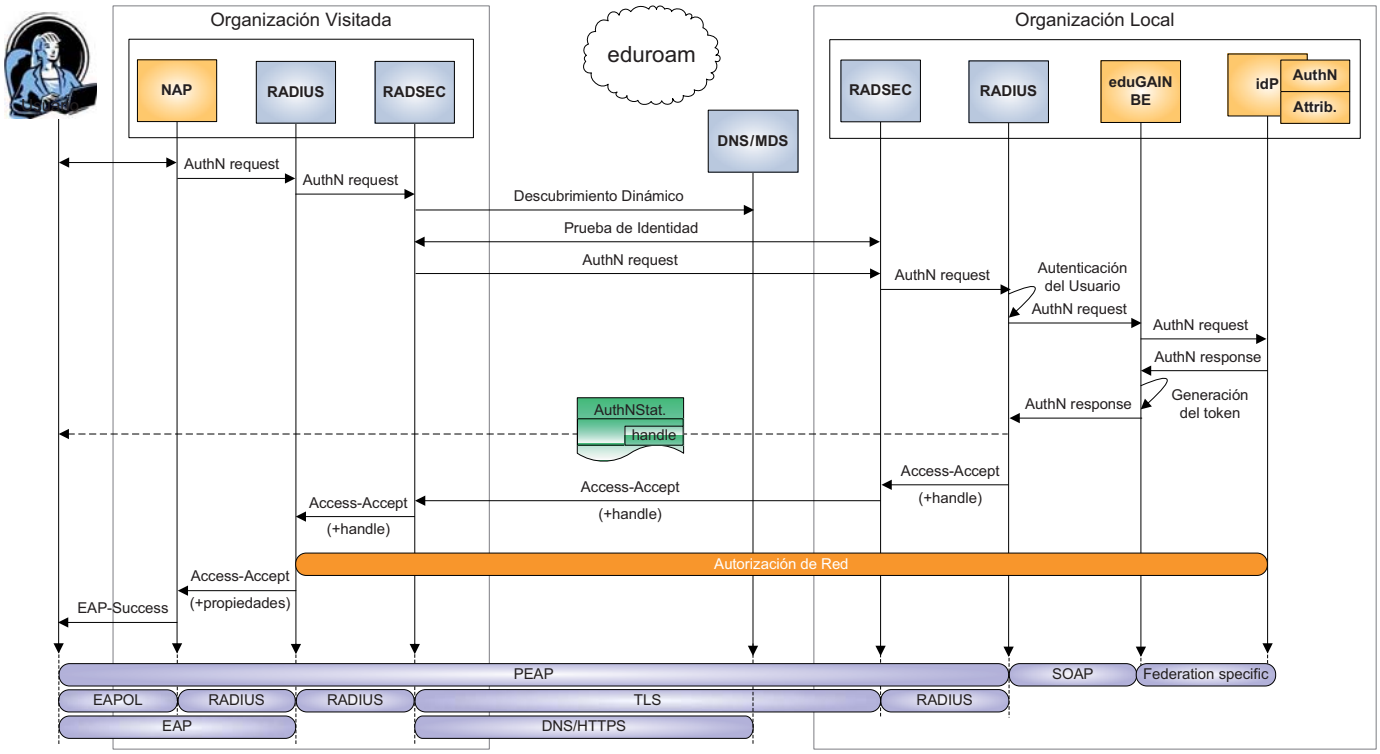


Fig. 2. Proceso de autenticación en DAMe haciendo uso de RadSec

puede hacer uso de sus propios métodos de autenticación y autorización. Para hacer uso de eduGAIN se deben desplegar los *Bridging Elements* (BE) correspondientes para la traducción de mensajes de autorización, autenticación y petición de atributos.

En el escenario de la Figura 1 mostramos dos organizaciones. Una es la organización local del usuario, que aloja su *Proveedor de Identidad* (IdP); por lo tanto, define como mínimo dos entidades, una autoridad de autenticación y una autoridad de atributos a la que solicitar información sobre dicho usuario. Además, es necesaria la existencia de un servidor RADIUS [5] conectado a eduroam para la fase de autenticación del usuario.

En la organización destino aparecen tanto un servidor RADIUS conectado a eduroam, igual que en la organización local, un BE de eduGAIN que se encargará de traducir los mensajes recibidos a formato interno, y una entidad llamada *Policy Decision Point* (PDP) que se encarga de tomar la decisión de autorización en base a los atributos recibidos de la organización local.

A. Autenticación de red

En DAMe esta fase se basa en eduroam para poder autenticar al usuario a nivel de red, utilizando para ello la jerarquía de servidores RADIUS ya establecida. La extensión que se añade sobre este proceso consiste en que el servidor de autenticación de la organización local pueda solicitar a su BE un token (eduToken), que recibirá el usuario, con el que podrá acceder a otros servicios mediante SSO.

Para evitar un posible robo del token por un usuario malintencionado, se debe proteger tanto el envío como el almacenamiento en el dispositivo del usuario.

B. Distribución del token SSO de autenticación

Para establecer un canal seguro entre el usuario y el servidor de autenticación, para el envío del token, podemos hacer uso de métodos EAP, como por ejemplo TTLS [6] o PEAP [7]. El token se almacena en el dispositivo del usuario de forma segura, para que luego pueda ser utilizado por los servicios a nivel de aplicación.

C. Fase de autorización

Una vez el usuario ha obtenido el token de autenticación, la organización visitada debe conocer las capacidades del usuario para decidir a qué servicios puede acceder, que características tendrá su conexión, etc.

Para ello se establece una segunda fase en la que se realiza una consulta de los atributos del usuario, obtenidos desde su organización local a través de eduGAIN, que serán enviados al PDP para tomar la decisión de autorizar o no al usuario en base a esos atributos.

DAMe se basa en el uso de tecnologías estándar, como SAML [8] o XACML [9]. Una descripción más detallada de este proceso puede encontrarse en [10].

III. RadSec

RadSec [2] ofrece una solución para implementar RADIUS haciendo uso de conexiones TCP/TLS, ayudando a que las transferencias de información de autenticación y autorización sean más eficientes y seguras.

En eduroam se hace uso de una jerarquía RADIUS tradicional, descrita anteriormente, donde todos los mensajes de autenticación y autorización recorren la jerarquía desde el servidor RADIUS remoto al local. Todo este intercambio, que se hace mediante UDP y usando un cifrado basado en secretos compartidos, se pretende cambiar, a nivel organizativo, por una conexión directa entre ambos servidores RADIUS y una arquitectura de clave pública para su seguridad.

La principal idea de RadSec, además de establecer seguridad y eficiencia en la comunicación, es evitar que se deba hacer uso de toda la jerarquía de eduroam para la comunicación entre la organización local y visitada (como queda representado en la Figura 2). Para ello, se establece un canal directo entre los servidores RadSec remoto y local, estableciendo todo el intercambio de autorización y autenticación a través de dicho canal.

Para que esta conexión pueda llevarse a cabo mediante TCP/TLS deben existir una serie de elementos indispensables entre ambos servidores RadSec:

- Certificado de clave pública para el servidor que soporta RadSec y su correspondiente clave privada.
- Autoridad de certificación (CA) emisora.
- Es necesario un modelo de confianza entre ambas organizaciones, por ejemplo mediante una jerarquía de certificación común o a través de confianza mutua.
- Servicios de validación y descubrimiento.

Finalmente, hay que tener en cuenta las características de autodescubrimiento que se plantean en RadSec, ya que la configuración estática de los servidores conocidos plantea un serio problema de escalabilidad [11]. Se han propuesto varias alternativas basadas en DNS [12], DNSSec [13] o publicación de metadatos, que serán descritas en la siguiente sección. Actualmente la solución más utilizada es FreeRADIUS [14], que no tiene aún soporte para RadSec, por lo que será necesario hacer uso de entidades proxy, como *radsecproxy* [16], para desplegar esta solución.

IV. Arquitectura y Despliegue

Entre las distintas posibilidades de integración disponibles nos encontramos con Radiator [15], un software stand-alone de RADIUS que lleva integrado RadSec de forma nativa. Otra opción, por la que finalmente hemos optado, es la instalación de servidores *radsecproxy* que actúen como proxys entre los servidores FreeRADIUS.

La arquitectura propuesta con RadSec quedaría como se puede ver en la Figura 2, donde los principales elementos de la arquitectura son:

- **Supplicant**: Software instalado en el dispositivo del usuario para conectarse a la red. También debe permitir la creación de un canal seguro entre el usuario y el servidor RADIUS de su organización local.
- **NAP**: Punto de acceso que provee al usuario de acceso a la red.
- **RADIUS Remoto**: Responsable de redirigir las peticiones de autenticación del usuario hacia el servidor RadSec de su organización.
- **RadSec Remoto**: Responsable de descubrir el servicio RadSec de la organización local del usuario y de enviar las peticiones recibidas del servidor RADIUS hacia éste.
- **Servidor DNS/MDS**: Para el descubrimiento de la organización local, será necesario definir el mecanismo a utilizar, DNS o MDS. Los requerimientos de estos servicios se describirán más adelante.
- **RadSec Local**: Servidor RadSec de la organización local que recibe las peticiones de conexión del usuario desde el servidor RadSec Remoto. Estas peticiones serán reenviadas al servidor RADIUS Local para que realice la autenticación del usuario.
- **RADIUS Local**: En base a la petición de autenticación recibida realiza la autenticación correspondiente y solicita al IdP, a través del BE Local, una sentencia SAML (eduToken) que será enviado al usuario y con el que podrá comenzar futuras sesiones de SSO.
- **BE Local**: Encargado de traducir la solicitud del servidor RADIUS Local a la tecnología utilizada por el IdP. Con la información recibida de éste, el BE Local creará el eduToken que le será enviado al usuario, en función de la información recibida del IdP.
- **IdP**: Entidad que maneja los atributos y las identidades de los usuarios, la cual recibe consultas de autenticación desde el BE Local. También recibe peticiones de atributos de los usuarios durante la fase de autorización. El IdP puede estar basado en diferentes tecnologías como Shibboleth [17], PAPI [18], etc.

La interacción entre las entidades añadidas para la integración de RadSec en DAMe queda reflejada en la Figura 2. A continuación, mostramos paso a paso el proceso completo que seguiría una interacción común de autenticación en DAMe con RadSec.

A. Autenticación

Inicialmente, el Supplicant del usuario hace una petición de acceso a la red al NAP y, como en un esquema eduroam común, éste envía un mensaje *Access-Request* al servidor RADIUS de la organización visitada.

En este punto, el servidor RADIUS Remoto, que tendrá configurado el servidor RadSec de su organización como proxy, le reenviará el mensaje mediante un mensaje RADIUS UDP común y será el proxy RadSec el encargado de transmitirlo a la organización local a través de un canal seguro entre ambas organizaciones.

El servidor RadSec Remoto procede con el descubrimiento del servidor RadSec de la organización local. El proceso de descubrimiento dinámico de entidades RadSec queda explicado en el siguiente subapartado. Una vez descubierto el servidor RadSec de la organización local, se procede a crear un canal seguro entre los dos servidores haciendo uso de TLS. A partir de este momento, todos los mensajes se intercambiarán de manera segura.

Una vez que el mensaje de autenticación llega al servidor RadSec Local, éste se encontrará configurado para retransmi-

tirlo al servidor RADIUS de su organización, que procederá con el proceso de autenticación.

Para dicha autenticación se sigue el funcionamiento común de DAMe, realizando una petición del token de autenticación (edutToken) al IdP local, a través del BE Local.

Una vez tomada la decisión de autenticación, se procede a enviar el mensaje *Access-Accept* desde la organización local a la visitada. En este caso, el servidor RADIUS Local, que tendrá configurado el servidor RadSec de su organización como proxy, le enviará el mensaje para que sea reenviado a través del canal seguro TLS. Una vez que el servidor RadSec Remoto recibe dicho mensaje, lo reenvía a su servidor RADIUS que será el encargado de hacerlo llegar al punto de acceso desde donde el usuario está tratando conectarse, y que le dará el permiso para poder acceder a la red.

B. Transmisión del token y fase de autorización

Una vez que el servidor RADIUS Local obtiene el token SSO, lo envía a través del canal seguro y éste se almacena en el dispositivo del usuario, tal y como se comentó anteriormente.

Respecto a la fase de autorización, no se ve afectada por la integración de RadSec, por lo que se sigue el proceso normal que se realiza en DAMe.

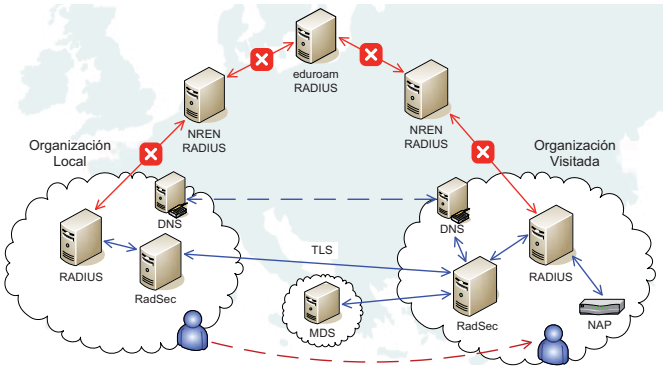


Fig. 3. Descubrimiento dinámico en RadSec

C. Autodescubrimiento

Una vez definida la integración de los servidores RadSec en DAMe, se debe plantear una solución al problema del descubrimiento de los puntos de autenticación del resto de organizaciones.

En nuestro caso estudiamos dos posibles vías: haciendo uso del servicio de descubrimiento de nombres DNS; y mediante el uso de un servicio común de metadatos (MDS) siguiendo la propuesta de eduGAIN.

1) Autodescubrimiento usando DNS: Para llevar a cabo este método, la organización local debe anunciar, mediante una entrada SRV en su servidor DNS, el servidor RadSec que estará esperando conexiones entrantes de otras organizaciones. El servidor RadSec Remoto hará una consulta DNS preguntando por el servicio *_radsec._tcp* dentro del dominio local, como se establece en [2].

Una vez conocido el servidor RadSec de la otra organización, se procede al establecimiento del canal seguro, haciendo uso de los certificados que tanto el servidor de la organización visitada como el de la local tienen. El uso de DNSSec en RadSec también se ha propuesto, aunque queda fuera del ámbito de este trabajo y se planteará como trabajo futuro.

En la Figura 3 puede verse un esquema de la interacción del DNS en el esquema de DAMe.

2) Autodescubrimiento usando eduGAIN MDS: En este otro caso se ha usado la estructura de metadatos ofrecida por *eduGAIN* para hacer el anuncio de servicios.

```
<md:EntityDescriptor ID="..." entityID="...">
  <md:IDPSSODescriptor ID="USTUTT-RADSEC">
    <md:SingleSignOnService Location="radsec(*)://ksat.rus.uni-stuttgart.de:2083"/>
  </md:IDPSSODescriptor>
  <md:Organization>
    <md:Extensions>
      <egmd:HLPattern egmd:MatchingAlgo="urn:..:metadata:exact">
        uni-stuttgart.de
      </egmd:HLPattern>
    </md:Extensions>
  </md:Organization>
</md:EntityDescriptor>
```

Listing 1. Ejemplo de fichero MDS

Será la organización local la que haga uso de un repositorio de metadatos para anunciar su servidor RadSec, y será el cliente (en nuestro caso el servidor RadSec Remoto) el encargado de preguntar a dicho repositorio por el servidor que desea localizar. Esta solución tiene algunas consideraciones de seguridad a ser tomadas en cuenta, como por ejemplo:

- El servidor remoto debe asegurarse que pregunta al MDS correcto, haciendo uso de los certificados correspondientes.
- El MDS debe validar la información que anuncia.

En el Listado 1 mostramos un ejemplo de la información almacenada en el MDS, que servirá para el descubrimiento del servicio de RadSec en la organización local.

En la Figura 3 se ve también la integración del servidor MDS para la obtención de la información necesaria por parte de la organización visitada.

V. Medidas de Rendimiento

En esta sección detallamos la viabilidad a la hora de introducir RadSec en la arquitectura actual de DAMe.

Elemento	Hardware	Software
Organización Visitada (Universidad de Murcia)		
Suppplicant	Pentium M 1.60 GHz 512MB RAM	WPASupplicant 0.6.3
NAP	Linksys WRT54G	-
RADIUS	AMD Opteron 246 1GB RAM	FreeRADIUS 1.1.3
RadSec		radsecproxy 1.3.1
Organización Local (Universidad de Stuttgart)		
DNS	Pentium Dual 2GHz 2GB RAM	BIND 9.4.2-P2.1
MDS		Apache 2.2/mod_ssl
RadSec		radsecproxy 1.3.1
RADIUS		FreeRADIUS 2.0.2
BE Local	Pentium IV 3GHz 1GB RAM	Tomcat 5.5.27
IdP		Shibboleth 1.3 OpenLDAP 2.3.28

TABLE I
ELEMENTOS HARDWARE Y SOFTWARE DEL TESTBED

Autenticación estándar (eduroam)	Autenticación estándar (eduroam + RadSec)	Autenticación extendida (eduroam + RadSec + eduToken)		
1498 / 753	1319	2082 / 2389		
		DNS/MDS	Autenticación Shibboleth	Obtener eduToken
		81 / 388	54	628

TABLE II
MEDIANA DE LOS TIEMPOS OBTENIDOS PARA LAS MEDIDAS DE RENDIMIENTO (MS)

Para ello, se ha utilizado la infraestructura desplegada en DAMe, incluyendo los servicios que tanto eduroam y eduGAIN proporcionan para, respectivamente, autenticar y autorizar a los usuarios de la federación. Como hemos comentado anteriormente, esta arquitectura se ha extendido añadiendo únicamente dos servidores RadSec (uno por organización).

Las pruebas realizadas se han llevado a cabo entre la Universidad de Murcia (organización visitada donde el usuario quiere conectarse a la red) y la Universidad de Stuttgart (organización local a la que el usuario pertenece).

En la Tabla I mostramos los detalles hardware y software de los elementos utilizados para desplegar el escenario explicado anteriormente (ver Figura 2). El sistema operativo utilizado en todos los componentes es una Ubuntu Kernel 2.6.24, a excepción del BE Local/IdP que es un Windows 2003 SP2.

Todas las muestras de tiempo obtenidas en esta sección, medidas en milisegundos, se han realizado ejecutando 105 veces la prueba correspondiente de forma secuencial.

Las pruebas que se han realizado, como se puede ver en la Tabla II (una prueba por columna), son las siguientes:

- Autenticación estándar/básica en eduroam.
- Autenticación estándar en eduroam a través de los servidores RadSec.
- Autenticación extendida en eduroam, a través de los servidores RadSec, incluyendo la generación del token SSO (eduToken) por parte del IdP de Stuttgart.

En este último caso hemos realizado dos pruebas diferentes, según el método de autodescubrimiento utilizado por el RadSec Remoto para localizar el servidor RadSec Local de la Universidad de Stuttgart: mediante DNS o MDS. Ambos servicios, como podemos ver en la Tabla I, han sido instalados en la organización local (Universidad de Stuttgart).

En la Tabla II se muestran las medianas de los tiempos obtenidos para cada una de las pruebas anteriores. En la primera columna se visualizan los tiempos para una autenticación estándar con eduroam. En este caso, hemos realizado dos pruebas:

- Estableciendo un canal PEAP con Stuttgart usando toda la jerarquía RADIUS de eduroam, compuesta por 5 servidores en total, obteniendo un tiempo de 1498ms. Viendo estos tiempos podemos ver cómo la utilización de RadSec (segunda columna) es ligeramente mejor, 1319ms frente a 1498ms, ya que se evita tener que recorrer toda la jerarquía RADIUS de eduroam.
- Estableciendo el canal PEAP directamente entre los dos servidores RADIUS de ambas universidades, obteniendo un tiempo de 753ms. Esta prueba se ha realizado para comprobar qué diferencia de tiempo hay entre utilizar RadSec o no, sin tener en cuenta la jerarquía RADIUS de eduroam, ya que RadSec no la utiliza. En este caso, podemos ver que la introducción de RadSec supone un incremento de 753ms a 1319ms; es decir, un 75% de sobrecarga.

Como primera conclusión de estos tiempos podemos afirmar que, aunque la introducción de RadSec supone un incremento considerable, los tiempos en DAMe son incluso ligeramente mejores (sobre el 13%, de 1498ms a 1319ms). Esta mejoría se debe a que ya no es necesario tener que recorrer toda la jerarquía RADIUS establecida en eduroam.

La última columna de la Tabla II muestra el tiempo obtenido en la autenticación eduroam, con soporte RadSec, incluyendo la generación del eduToken, como se realiza en DAMe. En esta prueba, los tiempos se dividen dependiendo del tipo de autodescubrimiento utilizado. Haciendo uso de un servicio DNS obtenemos un tiempo total de 2082ms, mientras que la utilización de MDS supone un tiempo de 2389ms. Podemos comprobar que ambos tiempos son asumibles por parte del usuario, aunque la diferencia entre ambos métodos de autodescubrimiento es un factor bastante significativo (de 81ms para el DNS a 388ms para el caso del MDS).

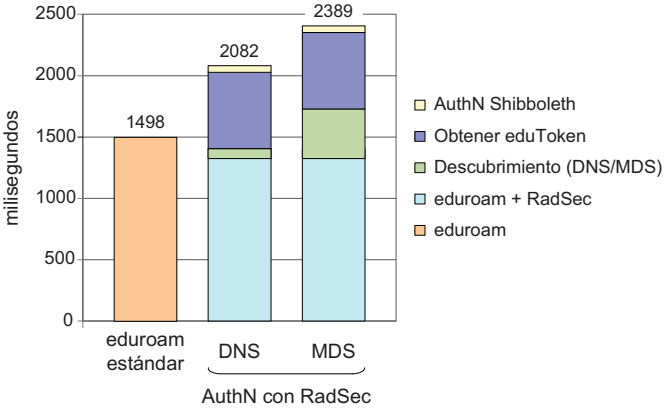


Fig. 4. Tiempos de autenticación con/sin RadSec

La Figura 4 muestra gráficamente los mismos tiempos que se incluyen en la Tabla II, para así poder compararlos de forma más intuitiva. En este caso podemos distinguir claramente el incremento que supone la utilización del autodescubrimiento por DNS y MDS.

Finalmente, todos los tiempos obtenidos utilizando el autodescubrimiento mediante DNS se ilustran en la Figura 5.

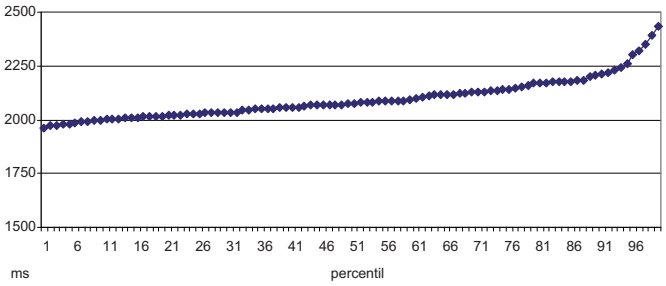


Fig. 5. Percentil de la autenticación con RadSec (DNS)

Todos los tiempos que hemos presentado en esta sección están relacionados con la fase de autenticación que se realiza a través de eduroam. Para el proceso completo de autenticación y autorización, como se establece en DAMe, faltaría incluir a estos tiempos la última fase de autorización por red. Este tiempo, también sobre DAMe, ha sido calculado en [19], obteniendo 531ms. Por tanto, si sumamos este tiempo al obtenido en esta sección para la fase de autenticación (2082ms, para el caso de DNS), tendríamos un tiempo total de 2613ms. Este tiempo sigue siendo asumible por el usuario final para obtener la conexión de red solicitada en la organización visitada.

VI. Trabajo Relacionado

No existen trabajos realizados sobre esta temática que incluyan una integración completa de RadSec en una arquitectura definida junto con medidas de rendimiento. Este trabajo se basa principalmente en la documentación generada en GÈANT [20][21], donde se presentan diferentes alternativas a la integración de RadSec en eduroam.

VII. Conclusiones y Trabajo Futuro

Este trabajo presenta una propuesta de integración de RadSec y DAMe sobre una red en producción como es eduroam, por lo que se ofrece una mejora significativa sobre los problemas de seguridad relacionados con RADIUS. Para comprobar la viabilidad de la propuesta se ha realizado un análisis del rendimiento sobre un escenario real, entre dos organizaciones conectadas a la federación. El análisis muestra que la integración de RadSec mejora los tiempos de comunicación sobre el uso tradicional de la jerarquía RADIUS, y que, además, la integración con DAMe supone unos tiempos asumibles por el usuario, teniendo en cuenta que además de mayor seguridad en las comunicaciones, podrá hacer uso de mecanismos avanzados de autorización y de SSO.

Como trabajo futuro se plantea la integración del escenario con mecanismos NEA para el control de acceso basado en atributos del dispositivo del usuario, y la integración del proceso de intercambio de atributos con RadSec.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto MULTIGIGABIT EUROPEAN ACADEMIC NETWORK (FP7-INFRASTRUCTURES-2009-1), y el programa de excelencia para grupos de investigación de la Fundación Séneca (04552/GERM/06).

References

[1] K. Wierenga et al. “Deliverable DJ5.1.4: Inter-NREN Roaming Architecture. Description and Development Items”. GN2 JRA5, GÉANT 2, Septiembre 2006.
[2] S. Winter, M. McCauley, S. Veenas and K. Wierenga. “TLS encryption for RADIUS”. IETF Internet-Draft 06, Marzo 2010.
[3] DAMe Project. Website: http://dame.inf.um.es
[4] D.R. López, J. Macías, M. Molina, J. Rauschenbach, A. Solberg and M. Stanica. “Deliverable DJ.5.2.3,2: Best Practices Guide - AAI Cookbook - Second Edition”. GN2 JRA5, GÉANT2, Marzo 2007.
[5] C. Rigney, S. Willens, A. Rubens and W. Simpson. “Remote Authentication Dial in User Service (RADIUS)”. RFC 2865, Junio 2000.
[6] P. Funk and S. Blake-Wilson. “EAP Tunneled TLS Authentication Protocol (EAP-TTLS)”. IETF Internet-Draft 05, Julio 2004.
[7] A. Palekar, D. Simon, J. Salowey, H. Zhou, G. Zorn and S. Josefsson. “Protected EAP protocol (PEAP) Version 2”. IETF Internet-Draft 10, Octubre 2004.
[8] E. Maler, P. Mishra and R. Mishra (Ed.). “Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1”. OASIS Standard, Septiembre 2003.
[9] T. Moses (Ed.). “eXtensible Access Control Markup Language (XACML) Version 2.0”. OASIS Standard, Febrero 2005.
[10] O. Cánovas, M. Sánchez, G. López, R. del Campo, S. Neinert, J. Rauschenbach and I. Thomson. “Deliverable DJ.5.3.2: Architecture for Unified SSO”. GN2 JRA5, GÈANT2, Mayo 2008.
[11] T. Lenggenhager et al. “Deliverable DJ5.4.1,2: Advanced Technologies Overview, Second Edition”. GN2 JRA5, GÈANT2, Febrero 2009.
[12] S. Winter and M. McCauley. “NAI-based Dynamic Peer Discovery for RADIUS over TLS and DTLS”. IETF Internet-Draft 02, Marzo 2010.
[13] DNS Extensions (dnsext), IETF Network Working Group. Website: http://datatracker.ietf.org/wg/dnsext/charter
[14] The FreeRADIUS Project. Website: http://freeradius.org
[15] Radiator Project. Website: http://www.open.com.au/radiator
[16] radsecproxy. Website: http://software.uninett.no/radsecproxy
[17] T. Scavo and S. Cantor. “Shibboleth Architecture: Technical Overview”. Internet2 Working Draft 02, Junio 2005.
[18] D.R. López and R. Castro-Rojo. “Ubiquitous Internet Access Control: The PAPI System”. Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA), pp.441-445, 2002.
[19] M. Sánchez, G. López, O. Cánovas, A.F. Gómez-Skarmeta. “Performance Analysis of a Cross-layer SSO Mechanism for a Roaming Infrastructure”. Journal of Network and Computer Applications, 32(4):808-823, Febrero 2009.
[20] S. Winter, T. Wolniewicz and I. Thomson. “Deliverable DJ3.1.1: RadSec Standardisation and Definition of eduroam Extensions”. GN3 JRA3, GÈANT3, Noviembre 2009.
[21] S. Winter, M. Milinovic and I. Thomson. “Deliverable DS5.4.1: Report on RadSec Integration”. GN2 SA5, GÈANT2, Julio 2008.

REDUCCIÓN DE LA REDUNDANCIA DE CIFRADO EN REDES BASADAS EN TCP/IP Y 802.11

Antonio Urbano Fullana[1], Josep Lluís Ferrer Gomila[2] y Magdalena Payeras Capellà[3]
Universitat Illes Balears,
Dept. Matemáticas e Informática,
Ctra. Valldemossa, Km 7,5. 07120, Palma
e-mail antonio.urbano@uib.es[1] jlferrer@uib.es[2] mpayeras@uib.es[3]

***Resumen*—Los servicios tradicionales sobre redes cableadas pueden operar, en la actualidad, sobre redes inalámbricas. A los mecanismos de seguridad existentes en las capas del modelo TCP/IP, se añaden los mecanismos de seguridad de capa MAC y por tanto se generan duplicidades, e incluso multiplicidades, en el cifrado de determninados octetos. El conocimiento de los mecanismos de cifrado utilizados en las diferentes capas y en particular de los octetos cifrados por cada uno de ellas, puede ayudar a elimar esta redundancia de cifrado. En este artículo analizamos los servicios de seguridad en la arquitectura TCP/IP sobre redes IEEE 802.11, propoponemos una solución que elimina la redundancia en el cifrado y, finalmente, cuantificamos el número de octetos cifrados en función de la longitud de los datos realizando una comparativa entre los octetos que se cifrarían al aplicar la propuesta realizada y los cifrados actualmente.**

Keywords: Wireless Security, Security Protocols, WEP, IPSEC, 802.11.

I. INTRODUCCIÓN.

En este artículo presentamos un escenario donde un usuario A desea acceder a un servidor S a través del protocolo HTTPS. El servidor pertenece a una empresa que permite el acceso a sus servidores a través de conexiones VPN (Virtual Private Network). El usuario A es un terminal inalámbrico que opera en una red 802.11 [1] y que implementa seguridad a nivel MAC. Los servicios de seguridad requeridos en nuestro escenario son confidencialidad de la información extremo a extremo, el establecimiento de un túnel IPSEC (IP Secure) para disponer de seguridad puerta a puerta en los paquetes a nivel IP y seguridad a nivel MAC en 802.11. Este terminal implementará SSL a nivel de tranporte, IPSEC a nivel IP y WEP a nivel MAC. La red 802.11 de nuestro escenario es una red en modo infraestructura gestionada por un punto de acceso (AP) que también implementa WEP a nivel MAC. Los servicios de seguridad incluyen cifrado de la información y, en este artículo, demostramos que determinados octetos son cifrados por los tres mecanismos indicados. Definimos la Profundidad de Cifrado (PC) como el número de veces que un campo de un protocolo de capa ha sido cifrado en su descenso por la pila TCP/IP [10] hasta la capa física.

Este artículo se organiza como sigue. En la sección II estudiamos el encapsulado de los protocolos de la pila TCP/IP y de los servicios de seguridad SSL, IPSEC y WEP. El análisis de los mecanismos de cifrado utilizados y de la redundancia de cifrado se realiza en la sección III. La solución presentada en la sección IV elimina la duplicidad de cifrado. En la sección V presentamos los resultados obtenidos al aplicar la solución presentada y cuantificamos la reducción del número de octetos cifrados.

II. TCP/IP Y LOS SERVICIOS DE SEGURIDAD.

TCP/IP constituye el estándar de comunicaciones entre dispositivos conectados a Internet. TCP/IP es una colección de protocolos estándar de la industria diseñada para intercomunicar grandes redes WAN (Wide Area Network). El escenario presentado en la sección I se implementa sobre la pila TCP/IP y utiliza los protocolos de capa y mecanismos de seguridad que se indican en la tabla I:

Resumen de protocolos y mecanismos de seguridad en el escenario descrito		
Capa	Protocolo	Mecanismo Seguridad
APLICACION	HTTP	–
TRANSPORTE	TCP	SSL
RED	IP	IPSEC
ENLACE	802.11	WEP
FISICA	802.11	–

Tabla I
RESUMEN PROTOCOLOS

A continuación introducimos estos protocolos detallando las sobrecargas introducidas por cada uno y los octetos afectados por los mecanismos de cifrado de capa.

II-A. HTTP

El protocolo HTTP/1.1 (Hypertext Transfer Protocol) [9] es el protocolo utilizado por el servicio World Wide Web (WWW). HTTP está orientado a transacciones donde el cliente especifica en la información enviada al servidor que acción quiere ejecutar. Definimos L como la suma de las

longitudes de los datos y la cabecera del protocolo HTTP en octetos.

II-B. SSL

SSL [2] proporciona integridad y confidencialidad a los datos de aplicación. Opera a nivel de transporte y su arquitectura, detallada en la figura 1, está formada por 4 subprotocolos. SSL transmite los datos utilizando la clave y el algoritmo de cifrado negociados al inicio de la conexión SSL entre cliente y servidor.

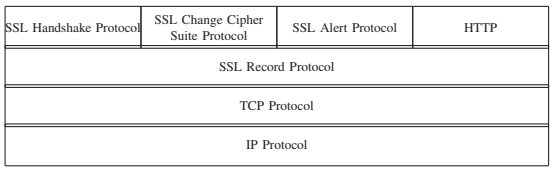


Figura 1. Arquitectura SSL.

En la capa de transporte, SSL divide los datos de la capa de aplicación en bloques de longitud máxima 16384 octetos sobre los que genera un hash. El campo de datos y el hash son cifrados utilizando el algoritmo acordado en la fase de negociación. La cabecera del protocolo SSL tiene una longitud de 5 octetos.

Con el objeto de proporcionar datos concretos, suponemos que el algoritmo hash utilizado es MD5, que genera un campo de resumen de 16 octetos y el algoritmo de cifrado negocidado es RC4-128. Con estos datos la longitud de datos de cifrado es de (L+16) y la información enviada al módulo TCP tiene una longitud de 5+L+16 = 21+L octetos (ver figura 2).

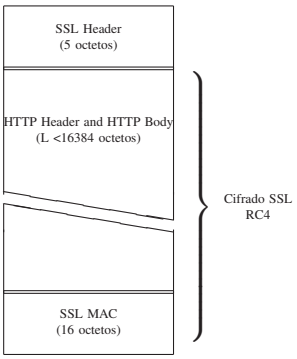


Figura 2. Estructura de datos de SSL Record Protocol

II-C. TCP

TCP [5] es el protocolo de capa de transporte. Divide los datos de aplicación en función del tamaño máximo del segmento que se define por el parámetro MSS (Maximum Segment Size). La cabecera TCP, sin considerar opciones, introduce un overhead de 20 bytes (ver figura 3), para formar un segmento de 20+5+L+16 = 41+L octetos. Consideremos que L es menor que el tamaño máximo de segmento.

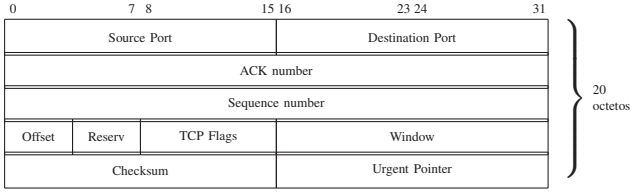


Figura 3. Cabecera TCP.

II-D. IP

El protocolo IP [4] opera en la capa de red y es el encargado de transportar los datos de un origen a un destino. El protocolo IPv4 introduce un overhead, debido a la cabecera (ver figura 4) de 20 octetos para formar un paquete de longitud 61+L octetos (ver figura 5).

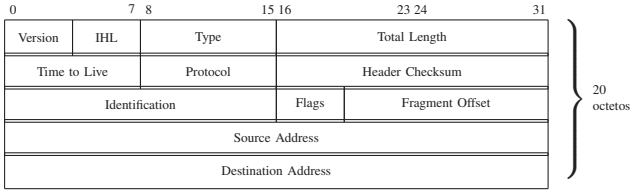


Figura 4. Cabecera IP versión 4

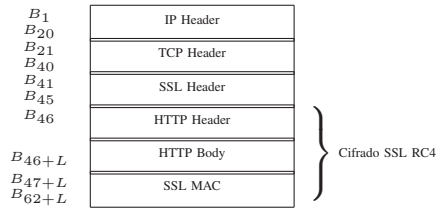


Figura 5. Formato paquete IP.

II-D1. IPSEC: IPSEC [6] es el protocolo que ofrece servicios de seguridad en la capa de red de la pila TCP/IP. IPSEC está formado por un conjunto de protocolos que operan individual o colectivamente para ofrecer mecanismos de seguridad:

1. IKE (Internet Key Exchange).
2. AH (Authentication Header).
3. ESP (Encapsulating Security Payload).

IKE [3] es el protocolo usado para establecer las asociaciones de seguridad en IPSEC. AH [7] proporciona integridad y autenticación al protocolo IP así como protección frente a ataques de repetición. ESP [8] además de los servicios ofrecidos por AH, proporciona confidencialidad al protocolo

IP.

IPSEC fue diseñado para ofrecer servicios de seguridad en dos modos de funcionamiento:

1. Modo túnel. La seguridad se establece puerta a puerta. En este modo, todo el paquete IP incluyendo cabeceras y datos es cifrado y autenticado por lo que es necesario generar un nuevo paquete IP. Este modo es utilizado para comunicaciones red a red u ordenador a ordenador sobre Internet.
2. Modo transporte. La seguridad se establece extremo a extremo. Son los dispositivos finales los que realizan todos los procesos asociados con la seguridad. En este modo sólo se cifra y autentica la carga útil del paquete. La cabecera IP no es modificada y la integridad de la información de las capas de transporte y aplicación se realiza mediante un hash de los datos.

En nuestro escenario consideramos que IPSEC implementa ESP en modo túnel para ofrecer confidencialidad en la cabecera IP. Para analizar el overhead introducido, suponemos que el algoritmo de cifrado utilizado es 3DES y que el algoritmo de autenticación es HMAC-MD5-96 (genera 12 octetos de información). IPSEC con el protocolo ESP en modo túnel y con los algoritmos indicados, añade un overhead de 52 octetos (8 octetos de cabecera ESP, 8 octetos de vector IV, 12 octetos de autenticación y 4 octetos del campo ESP trailer, más una nueva cabecera IP de 20 octetos) (ver figura 6). La longitud total del nuevo paquete con seguridad IPSEC es de 52+61+L = 113+L octetos.

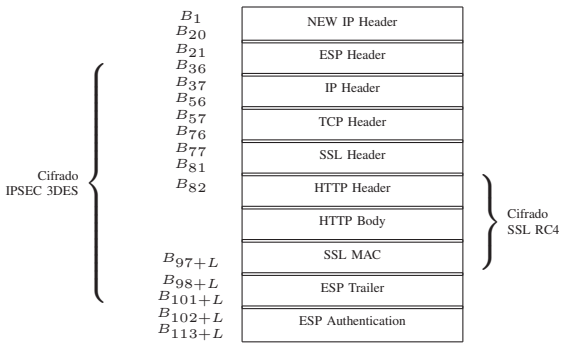


Figura 6. Formato paquete IPSEC, ESP modo túnel.

II-E. Nivel LLC

El nivel LLC añade 8 octetos de cabecera que sumados a los octetos del paquete IP con IPSEC suman 121+L octetos.

II-F. 802.11

El estándar 802.11 define las capas física y MAC, para comunicaciones en redes locales inalámbricas. El estándar ha evolucionado desde la versión 802.11a, que opera en la banda de 5Ghz a velocidades de hasta 54 Mbit/s, o la versión 802.11b, que opera en la banda de 2,4Ghz a velocidades de hasta 11 Mbit/s, hasta la versión 802.11n que implementa la nueva tecnología MIMO a velocidades de hasta 600 Mbit/s. El overhead introducido por 802.11 en la comunicación entre una estación y el punto de acceso es de 28 octetos y corresponde a la cabecera MAC y al campo FCS final. La longitud total de la trama 802.11 es de 155+L octetos, siendo 155+L menor que la longitud máxima de trama (2312 octetos).

II-F1. WEP: El estándard IEEE 802.11 [1] define WEP como mecanismo opcional de cifrado de capa MAC. WEP ofrece privacidad de datos a nivel MAC en una red inalámbrica y se basa en el algoritmo de cifrado RC4 y en sus variantes de clave de 40 bits (WEP-40), a la que se le añade un vector de inicialización (IV) de 24 bits para formar la clave RC4. Esto conforma el estándard WEP de 64 bits. La versión WEP de 128 bits utiliza una clave de RC4 de 104 bits (WEP-104) a la que se le añaden los 24 bits del vector IV. El overhead adicional a la cabecera 802.11 introducido por WEP es de 8 octetos que corresponden a los campos vector IV (4 octetos) y al campo ICV (4 octetos) correspondiente a la aplicación del CRC-32 del algoritmo WEP sobre los datos.

III. Análisis de los servicios de seguridad

En la sección anterior hemos analizado el encapsulado de los datos de aplicación hasta formar la trama 802.11 incluyendo los servicios de seguridad de cada capa (SSL, IPSEC y WEP) y los protocolos HTTP, TCP e IP. A continuación analizamos el impacto del cifrado realizado por cada uno de estos mecanismos de seguridad de capa sobre la información entregada por la capa superior.

El primer cifrado que se realiza es el que implementa SSL y se realiza sobre los campos de datos y hash del protocolo SSL Record con una longitud total de 16+L octetos (Ver figura 2). La PC (definida en la sección I) para estos campos es igual a 1.

Las capas TCP e IP añaden sus respectivas cabeceras al segmento entregado por SSL. IPSEC recibe el paquete IP de longitud 61+L octetos y el protocolo de seguridad ESP cifra el paquete IP y el campo ESP Trailer. El paquete IP original contiene los datos introducidos por la capa de aplicación y que han sido cifrados por SSL, con una longitud de 16+L octetos y que vuelven a ser cifrados por IPSEC, modificando su profundidad de cifrado a 2. Los campos de cabecera IP (20 octetos), cabecera TCP (20 octetos), cabecera SSL (5 octetos) y ESP Trailer (4 octetos) no han sido cifrados en capas superiores por lo que su PC será igual a 1 (ver figura 6).

El paquete IP junto a la cabecera LLC forman los datos a nivel 802.11 con una longitud de 121+L octetos y sobre los que se aplica el cifrado WEP. Los campos cifrados por IPSEC aumentan su PC por lo que a nivel MAC tendremos que la cabecera IP (20 octetos), la cabecera TCP (20 octetos),

la cabecera SSL (5 octetos) y el campo ESP Trailer (4 octetos) tienen ahora una PC igual a 2. Por otra parte el campo de datos de aplicación y el campo hash SSL (16+L octetos) vuelven a cifrarse por lo que su PC actual es igual a 3. En esta capa también se cifran los campos de cabecera LLC (8 octetos), la nueva cabecera IP del protocolo IPSEC (20 octetos), la cabecera ESP (16 octetos), el campo de autenticación del protocolo ESP, ESP AUTH (12 octetos) y el campo de CRC-32 de los datos ICV (4 octetos). Estos campos tienen una PC igual a 1.

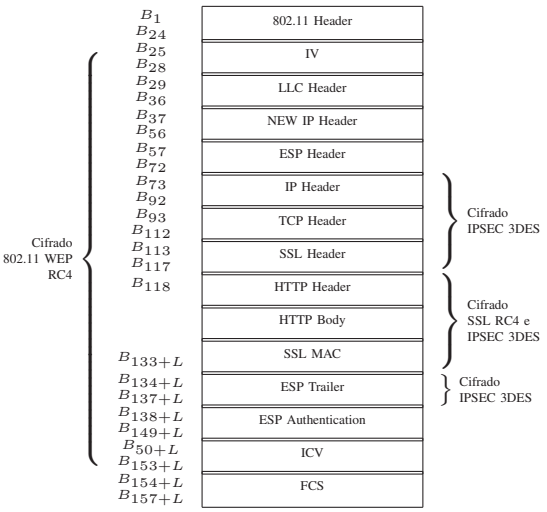


Figura 7. Formato de la trama 802.11.

La figura 7 describe la trama 802.11 con los mecanismos de cifrado aplicados y la tabla II resume la profundidad de cifrado de todos los campos de los protocolos descritos.

IV. Modificación en el cifrado de la información.

A continuación proponemos una solución para eliminar la redundancia de cifrado sobre los campos con una profundidad de cifrado superior a 1. El criterio adoptado en esta solución es que un campo que deba ser cifrado por el servicio de seguridad de capa lo será si su profundidad de cifrado es igual a 0, es decir, si no ha sido cifrado en capas superiores o es un campo introducido por el protocolo de capa. Este criterio permite mantener la seguridad de capas superiores al tiempo que elimina operaciones de cifrado en capas inferiores.

A continuación detallamos nuestra solución en cada uno de los mecanismos de seguridad de capa (SSL, IPSEC y WEP) y que se recogen en la figura 8.

IV-A. Modificaciones SSL.

Todos los datos de la capa de aplicación que recibe SSL tienen una PC igual a 0. Según el criterio adoptado, nuestra solución mantiene el cifrado SSL sobre el campo de datos de aplicación (L octetos) y sobre el campo SSL hash (16 octetos). La PC para estos campos es igual a 1.

IV-B. Modificaciones IPSEC

IPSEC recibe dos campos con PC igual a 1, que son, datos de aplicación (L octetos) y SSL hash (16 octetos). A estos campos no se les aplicará cifrado. Por otra parte los campos cabecera SSL (5 octetos), ESP Trailer (4 octetos), cabecera IP (20 octetos) y cabecera TCP (20 octetos) tienen una PC igual a 0, por lo que según el criterio indicado, IPSEC aplicará el cifrado y modificará la PC a 1.

IV-C. Modificaciones en seguridad WEP

Con la solución propuesta los campos cifrados por capas superiores que tienen una PC igual a 1 son los datos de aplicación (L octetos), SSL hash (16 octetos), la cabecera SSL (5 octetos), ESP Trailer (4 octetos), cabecera IP (20 octetos) y la cabecera TCP (20 octetos). Estos campos, aplicando el criterio de diseño propuesto no se cifran. El resto de campos que sí cifra WEP y que tienen PC igual a 0 son cabecera LLC (8 octetos), nueva cabecera IP (20 octetos), cabecera ESP (16 octetos), y el campo ESP Auth (12 octetos).

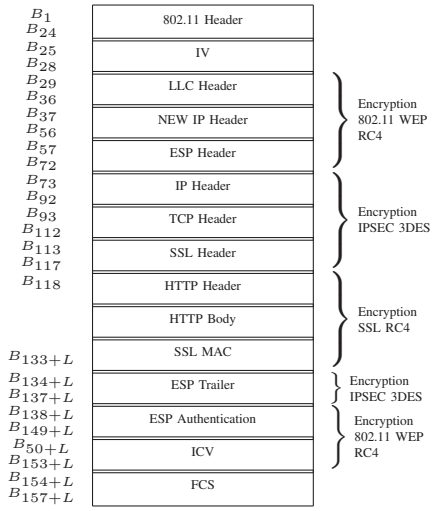


Figura 8. Formato trama 802.11 con modificación de cifrado.

IV-D. Resumen de los cambios en la aplicación de los mecanismos de cifrado.

Nuestra solución mantiene el cifrado SSL en la capa de transporte y suprime el cifrado de determinados campos a nivel IP (IPSEC) y a nivel MAC (WEP). En la tabla III se resume una comparativa del número de octetos total cifrados por los servicios de seguridad de capa antes y después de aplicar la solución presentada y referenciados a una longitud de datos de aplicación L.

Resumen de profundidad de cifrado				
Capa	Protocolo	Campo	Profundidad de cifrado	Mecanismos de cifrado aplicados.
APLICACION	HTTP	HTTP HEADER	3	SSL - IPSEC - WEP
APLICACION	HTTP	HTTP BODY	3	SSL - IPSEC - WEP
TRANSPORTE	SSL	SSL HEADER	2	IPSEC - WEP
TRANSPORTE	SSL	SSL HASH	3	SSL - IPSEC - WEP
TRANSPORTE	TCP	TCP HEADER	2	IPSEC - WEP
RED	IP	IP HEADER	2	IPSEC - WEP
RED	IPSEC	NEW IP HEADER	1	WEP
RED	IPSEC	ESP HEADER	1	WEP
RED	IPSEC	ESP TRAILER	2	IPSEC - WEP
RED	IPSEC	ESP AUTH	1	WEP
ENLACE	LLC	LLC HEADER	1	WEP
MAC	WEP	ICV	1	WEP

Tabla II
PROFUNDIDAD DE CIFRADO

Longitudes de cifrado (octetos)				
Capa	Cifrado	Longitud de datos	Cifrado Original	Nuevo Cifrado
TRANSPORTE	SSL (RC4)	41+L	16+L	16+L
IP	IPSEC-tunel (3DES)	113+L	65+L	49
MAC	WEP-128	157+L	125+L	60

Tabla III
VARIACIÓN DE LA LONGITUD CIFRADO

Longitudes de cifrado (octetos)									
Capa	Cifrado	Longitud Original L=1	Nueva longitud L=1	Longitud Original L=200	Nueva longitud L=200	Longitud Original L=800	Nueva longitud L=800	Longitud Original L=1300	Nueva longitud L=1300
TRANSPORTE	SSL (RC4)	17	17	216	216	816	816	1316	1316
IP	IPSEC-tunel (DES)	66	49	265	49	865	49	1365	49
MAC	WEP-128	126	60	325	60	925	60	1425	60
SUMA octetos	–	209	126	807	325	2607	925	4107	1425

Tabla IV
LONGITUD CIFRADO VARIANDO L

Destacamos que en la solución presentada, IPSEC y WEP cifrarán datos con longitud fija de 49 y 60 octetos independientemente de la longitud de los datos de aplicación, siempre que a niveles IP y transporte se implementen IPSEC ESP en modo túnel y SSL respectivamente.

En la tabla IV se muestra la longitud total de octetos cifrados por los mecanismos de seguridad en función de la longitud L. En nuestro escenario, para una longitud L y aplicando N mecanismos de cifrado, la estructura tradicional de seguridad cifra ((NxL)+209 octetos) con N=3. Al aplicar la solución propuesta, los mecanismos de seguridad cifrarán (L+126) octetos. La reducción de los octetos cifrados es del 39,71 % para una longitud L=1 octetos, de un 64,5 % para L=800 octetos y de un 65,4 % para L=1400 octetos. El límite teórico de reducción del número de octetos cifrados, que correponde a una longotud L suficientemente grande con la que podemos despreciar el cifrado de las cabeceras, es del 66,67 %.

V. RESULTADOS.

Hemos cuantificado que para una longitud de datos L, la arquitectura TCP/IP tradicional cifrará (NxL)+K octetos, siendo N el número de mecanismos de cifrado aplicados, L la longitud de datos y K una variable que dependen del overhead de los mecanismos de cifrado utilizados en el cifrado original. Como alternativa a esta redundancia de cifrado, hemos propuesto una solución que reduce el número de octetos cifrados a L+K', siendo L la longitud de datos y K' una variable que depende del overhead de los mecanismos de cifrado utilizados al aplicar nuestra solución. La tabla V detalla los valores de N, K y K' en función de los mecanismos de cifrado utilizados y la figura 9 representa las rectas que se obtienen al variar el valor de L. En las gráficas se indican los resultados correspondientes al cifrado sin aplicar nuestra solución sin referencia y referenciados como "CL" los resultados obtenidos al aplicar nuestra propuesta.

Longitudes de cifrado					
Cifrado	N	K	K'	(NxL)+K	L+K'
SSL+IPSEC+WEP	3	206	125	3L+206	L+125
IPSEC+WEP	2	148	104	2L+148	L+104
SSL+WEP	2	89	73	2L+89	L+73

Tabla V
VALORES DE N, K Y K'.

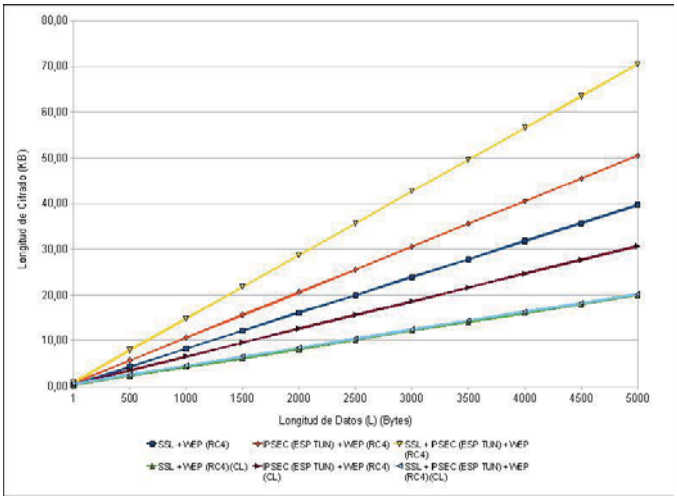


Figura 9. Longitud de Cifrado

El escenario presentado en el apartado I implementa SSL, IPSEC y WEP por lo que las expresiones que cuantifican los octetos cifrados son 3L+206 y L+125. A partir de estas expresiones obtenemos la figura 10 que muestra la variación del número de octetos cifrados por los mecanismos de capa en función de la longitud de los datos de aplicación (L) y compara nuestra solución con la situación de cifrado sin aplicar la solución presentada.

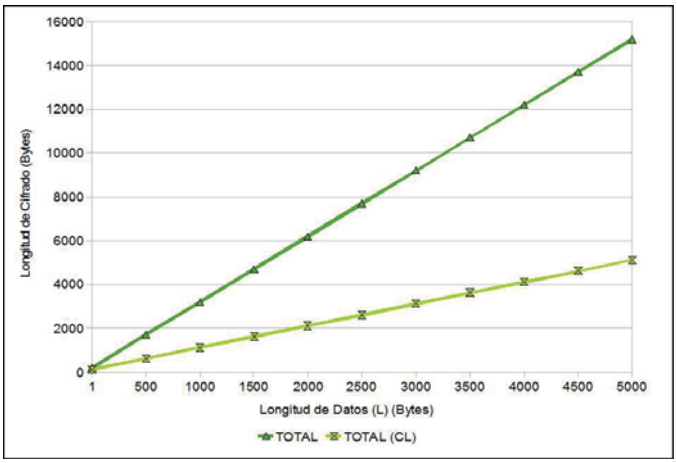


Figura 10. Número total de octetos cifrados.

VI. CONCLUSIONES Y TRABAJO FUTURO.

En este artículo hemos demostrado que al aplicar simultáneamente mecanismos de cifrado en diferentes capas se producen duplicidades, e incluso multiplicidades, en el cifrado de octetos. Hemos calculado que la longitud de cifrado, al aplicar mecanismos de cifrado en las capas TCP, IP y MAC, es (NxL)+K. Hemos propuesto una solución que reduce el cifrado a un orden L+K'.

Nuestra solución requiere que las capas dispongan de información del cifrado realizado en las otras capas. Este problema requiere la aplicación del diseño Cross-Layer en el que las distintas capas intercambien información relativa a las operaciones de cifrado que realizan. La línea de trabajo a seguir consiste en el diseño de un algoritmo Cross-Layer que implemente la solución propuesta. Otro aspecto importante es la cuantificación del impacto del cifrado en términos de energía y rendimiento global y los beneficios que se obtienen al aplicar una solución que reduzca la cantidad de octetos cifrados. Debemos considerar otros servicios de seguridad y mecanismos de cifrado de capa con el objeto de cuantificar la longitud de datos cifrados y los costes de energía en los terminales inalámbricos y el rendimiento en el punto de acceso.

VII. GRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el MEC y FEDER bajo los proyectos: "Seguridad en la Contratación Electrónica basada en Servicios Web"(CICYT TSI2007-62986) y ARES "Grupo de Investigación Avanzada en Seguridad y Privacidad de la Información"(Consolider - Ingenio CSD2007-004).

REFERENCIAS

[1] IEEE Std 802.11-1997 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.
[2] A.O. Freier, P. Karlton, P.C. Kocher. Transport Layer Security Working Group. *The SSL Protocol, V3.0. Internet Draft*, November 1996.
[3] D. Carrel D. Harkins. *RFC2409:The Internet Key Exchange (IKE)*. IETF. Network Working Group, November 1998.
[4] IETF. Network Working Group. *RFC791: Internet Protocol*, September 1991.
[5] IETF. Network Working Group. *RFC793: Transmission Control Protocol*, September 1991.
[6] S. Kent and R. Atkinson. *RFC2401: Security Architecture for the Internet Protocol*. IETF. Network Working Group, November 1998.
[7] S. Kent and R. Atkinson. *RFC2402: IP Authentication Header*. IETF. Network Working Group, November 1998.
[8] S. Kent and R. Atkinson. *RFC2406: IP Encapsulating Security Payload (ESP)*. IETF. Network Working Group, November 1998.
[9] R. Fielding H. Frystyk R. Fielding, J. Gettys and T. Berners-Lee. *RFC2068: Hypertext Transfer Protocol – HTTP/1.1*. IETF. Network Working Group, January 1997.
[10] T. J. Socolofsky and C.J. Kale. *RFC1180: TCP/IP tutorial*. IETF. Network Working Group, January 1991.

Modelo de Calidad para la Seguridad en Productos Software

Abel Enrique Fornaris
Grupo GSyA. Dep. de Tecnologías
y Sistemas de Información
Universidad de Castilla-La Mancha
Email: Fornaris@gmail.com

Luís Enrique Sánchez
Departamento de I+D+i
SICAMAN Nuevas Tecnologías
Juan José Rodrigo, 4. Tomelloso
Ciudad Real, España
Email: LESanchez@sicaman-nt.com

Eduardo Fernández-Medina
Grupo GSyA. Dep. de Tecnologías
y Sistemas de Información
Universidad de Castilla-La Mancha
Email: Eduardo.FdezMedina@uclm.es

***Resumen*—La importancia de tener en cuenta la seguridad como requisito no funcional en el éxito de un producto software (PS) es un hecho cada día más notable. Existen varios estándares y aproximaciones en la comunidad científica a la definición de la seguridad como elemento de calidad de un PS, sin embargo existen diferencias y falta de cohesión entre ellas. En este trabajo se estudian las propuestas más importantes en este sentido con el objetivo final de proponer un modelo de calidad integrador para la seguridad en el ámbito de los PS.**

***Palabras Clave*—seguridad; modelo calidad seguridad; seguridad producto software**

I. Introducción

La seguridad es un requisito no funcional que tiene una repercusión extraordinaria en la calidad de los productos software. De hecho, la seguridad informática ha sido un campo que ha crecido enormemente desde los años 70, dando lugar a una gran cantidad de técnicas, modelos, protocolos, etc., que han venido acompañados también de una actividad muy pronunciada por parte de las organizaciones internacionales de normalización y certificación. Tanto es así, que como se indica en [1], se pueden encontrar numerosas organizaciones internacionales de estandarización que han producido una compleja estructura de estándares relativos a temáticas relacionadas con la seguridad informática, que cambian y se actualizan con mucha frecuencia.

Existen numerosas definiciones de seguridad. Lo habitual es que todas ellas definan la seguridad en términos de otros conceptos relacionados. Por ejemplo, una definición tradicional es la de [2], que la define como la ”protección de información procesada por un computador frente a consultas no autorizadas, modificaciones inapropiadas o la falta de disponibilidad de un servicio en un momento dado”. Otra definición clásica es la ofrecida por [3], que considera la seguridad como un sub-factor de la calidad del software, y la define como ”la capacidad de los productos software para proteger los datos y la información para que personas o sistemas no autorizados no puedan leerla o modificarla y para que el acceso no sea rechazado a personal autorizado”. En ambas definiciones están presentes los conceptos de confidencialidad, integridad y disponibilidad. Sin embargo, hay otras definiciones algo más recientes que consideran además, otras propiedades importantes, como son la autenticación, el no repudio y la autorización y control de acceso [4]. Aunque la seguridad se puede interpretar como un aspecto estrictamente técnico, hay autores que piensan que es mucho más que eso, teniendo por el contrario una dimensión estratégica, resultando uno de los criterios más importantes en el gobierno de las TIC [5].

Sin embargo, aunque también se han desarrollado ampliamente en las últimas décadas las técnicas y metodologías propias de la ingeniería del software, éstas no han considerado la seguridad como un aspecto importante del desarrollo, dejando que la construcción metodológica del software se centre fundamentalmente en los requisitos funcionales, y algunos otros requisitos no funcionales, y relegando los requisitos de seguridad a un momento tardío en el proceso de desarrollo de software. Algunas propuestas interesantes que tratan la seguridad, aunque de manera parcial y sin ofrecer un claro seguimiento de esos aspectos de seguridad a lo largo del proceso de desarrollo incluyen [6–13].

Por lo tanto, se hace necesaria la creación de un modelo de seguridad (como componente de calidad) que claramente identifique una taxonomía de requisitos de seguridad que puedan ser identificados, modelados e implementados, junto al resto de requisitos, tanto funcionales como no funcionales.

El objetivo de esta propuesta es analizar los modelos existentes que definan la seguridad y sus componentes como aspectos que inciden en la calidad de los productos software, y construir un modelo unificado, completo y detallado que permita evaluar y mejorar este aspecto del desarrollo que resulta crítico para el éxito de muchos productos software. Este modelo, no incluirá aspectos relativos a técnicas de seguridad, amenazas de seguridad, políticas de seguridad, ataques de seguridad, etc., sino que fundamentalmente especificará las características de seguridad que nos puedan interesar de un producto software.

Para ello, se ha organizado el resto del documento del siguiente modo: En la Sección II se presentan algunos de los modelos de seguridad más relevantes. Posteriormente, en la Sección III se analizan paso a paso y desde un punto de vista integrador todas las características que proponen los trabajos estudiados en la sección anterior y que darán lugar al modelo

de seguridad que se propone en la Sección IV. La Sección V refleja las conclusiones obtenidas de la evolución y desarrollo de la propuesta.

II. Modelos de Seguridad Relevantes

En esta sección se describirán los aspectos más importantes de los modelos de seguridad definidos en los principales estándares más aceptados sobre calidad del software y seguridad.

A. ISO/IEC 9126

Uno de los estándares con mayor reconocimiento para evaluar la calidad de los productos software es el ISO/IEC 9126 [3]. Este estándar define tres tipos de calidad: la calidad interna, la calidad externa y la calidad en uso. El estándar define un modelo de calidad de productos software (tanto para calidad interna como externa) en términos de un conjunto de características (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad). Este modelo, que ya cuenta con más de una década desde su creación, otorga muy poca importancia a la seguridad como factor de calidad de los productos software.

En cuanto a la calidad en uso, el estándar la define como la capacidad de un producto software para permitir a ciertos usuarios conseguir sus objetivos con eficacia, productividad, seguridad y satisfacción en un contexto de uso específico. En este caso, también se hace una mención explícita a la seguridad, pero relativa al término inglés safety.

Un resumen de dicho modelo puede verse en la Tabla I , donde la seguridad no es una característica del mismo. Sin embargo, sí aparece definida como una sub-característica de la funcionalidad.

Tabla I
Modelo de Seguridad de la ISO/IEC 9126

Característica	Sub-característica	Sub-sub-característica
Modelo de Calidad Interna y Externa		
Funcionalidad	Seguridad	Confidencialidad
		Integridad
		Disponibilidad
Modelo de Calidad en Uso		
Seguridad (safety)		

B. SQuaRE - ISO/IEC 25010

SQuaRE (Software Product Quality Requirements and Evaluation - Requisitos y Evaluación de la Calidad de Productos Software) es una familia de estándares, que tienen como origen el estándar ISO/IEC 9126, y que define la calidad de un producto software como el grado en el que dicho producto satisface las necesidades implícitas y explícitas de sus diferentes usuarios [14]. Este estándar está en proceso de creación, y todavía no están disponibles documentos definitivos, aunque ya se puede intuir con cierta precisión sus aspectos más destacables.

Este estándar considera actualmente tres modelos de calidad: El modelo de calidad de productos software, el modelo de calidad en uso, y el modelo de calidad de datos. Cada uno de estos modelos es definido en términos de un conjunto de propiedades o características. Así, el modelo de calidad de productos software, considera entre sus características la seguridad y la fiabilidad, como destacables. En este modelo, el concepto de disponibilidad se incluye como una sub-característica de la fiabilidad. Por otro lado, el modelo de calidad en uso viene caracterizado por tres aspectos, que son la usabilidad, la flexibilidad y la seguridad (safety), este último de interés, con sub-características asociadas. Por último, el modelo de calidad de datos define un conjunto de características de las cuales son de interés las de confidencialidad y disponibilidad.

Este estándar, a pesar de no estar centrado en la seguridad (como otros que se analizan a continuación) ya ofrece un conjunto más completo de propiedades de seguridad, como se puede ver resumido en la Tabla II, y le otorga a la seguridad un protagonismo como aspecto de calidad que no era reconocido en las versiones anteriores del estándar del que parte SQuaRE, al pasar a estar presente como una característica más en el modelo.

Tabla II
Modelo de Seguridad de la ISO/IEC 25010

Característica	Sub-característica
Modelo de Calidad de Productos Software	
Seguridad	Confidencialidad
	Integridad
	No repudio
	Responsabilidad
	Autenticidad
	Conformidad
Modelo de Calidad en Uso	
Seguridad (safety)	Daño comercial
	Seguridad y salud del operador
	Seguridad y salud pública
	Daño medioambiental
	Conformidad de seguridad (safety)
Modelo de Calidad de Datos	
Confidencialidad	
Disponibilidad	

C. Modelo de Firesmith

Un modelo propuesto por Firesmith [15], lejos de la vorágine de las organizaciones de estandarización, y más en un contexto científico, propone un modelo de la seguridad, como característica de la calidad del software, compuesto por un conjunto de sub-características que a su vez se dividen en sub-características, con la organización que se representa en la Tabla III.

Cada elemento de la propuesta está debidamente definido, excepto las dos últimas sub-características de seguridad.

D. ISO/IEC 15408 o Common Criteria

Este estándar [16] no propone un modelo de seguridad, sino que propone un marco de trabajo para la evaluación

Tabla III
MODELO DE SEGURIDAD DE FIRESMITH

Característica	Sub-característica	Sub-sub-característica
Seguridad	Control de acceso	Identificación
		Autenticación
		Autorización
	Detección de ataques	
	No repudio	
	Integridad	Integridad de datos
		Integridad de hardware
		Integridad personal
		Integridad de software
	Auditoría de seguridad	
	Protección física	
	Privacidad	Anonimato
		Confidencialidad
	Recuperación	
	Continuidad	

de la seguridad de productos software y sistemas de información. El mismo no considera la seguridad como un requisito no funcional, sino todo lo contrario, define un conjunto de clases de componentes o requisitos funcionales de seguridad y también un conjunto de componentes de garantía de seguridad organizados en varios niveles de exigencia. Para este trabajo nos vamos a concentrar en las clases de requisitos funcionales, que se dividen en familias funcionales de seguridad, y que a su vez se dividen en componentes. Presentaremos a continuación algunas de las clases y familias, exponiendo un nivel adecuado para acercarnos al problema que nos atañe.

- Clase comunicaciones: Incluye familias de No Repudio de origen y recepción.
- Clase protección de datos de usuarios: Incluye familias de políticas y funciones de control de acceso; autenticación, importación/exportación, recuperación y transferencia interna de datos; políticas de protección de control de flujo, confidencialidad e integridad de datos en tránsito, protección e integridad de información residual y datos almacenados.
- Clase de identificación y autenticación: Familias de autenticación, identificación y enlace entre usuarios; atributos de usuario.
- Clase de privacidad: Incluye familias de anonimato, pseudoanonimato, enlace entre usuario y acciones, observación por parte de otros usuarios.
- Clase de protección de seguridad de un elemento: Familias de disponibilidad, confidencialidad e integridad de datos exportados; protección física, recuperación confiable, repetición de borrado.
- Clase de acceso a elementos: Familias de limitación, establecimiento, terminación y bloqueo de sesiones, historia de accesos.
- Clase de canales y caminos seguros.

E. MAGERIT versión 2

Este documento [17], elaborado por el Consejo Superior de Administración Electrónica de las Administraciones Públicas de España, presenta una metodología de análisis y gestión de riesgos de los sistemas de información, que resulta un documento de referencia tanto a nivel nacional como internacional (metodología oficialmente reconocida por la OTAN [18] y por la OCDE [19]) para la gestión de riesgos, y que es conforme a varias normas internacionales de gestión de la seguridad como es el caso de la ISO/IEC 13335 [20].

Esta metodología no ofrece un modelo de seguridad, pero si identifica una serie de aspectos que son cruciales para cumplir el objetivo de este trabajo. En primer lugar, MAGERIT identifica un conjunto de dimensiones de valoración, que son características o atributos que hacen valioso un activo, es decir, son facetas (sub-características) de seguridad que conviene proteger, en relación con los activos o elementos que constituyen valor dentro del contexto de las tecnologías de la información. En concreto, MAGERIT define las 7 dimensiones de valoración que se resumen en la Tabla IV, del siguiente modo:

Tabla IV
DIMENSIONES DE VALORACIÓN DE MAGERIT

Característica	Sub-característica
Dimensiones de Valoración de Activos	Disponibilidad
	Integridad de datos
	Confidencialidad de los datos
	Autenticidad de los usuarios del servicio
	Autenticidad del origen de los datos
	Trazabilidad del servicio
	Trazabilidad de los datos

Un segundo aspecto, considerado por MAGERIT, y que resulta especialmente interesante para el trabajo elaborado en este informe es el relativo a las amenazas. Las amenazas representan el impulso que da lugar a requisitos de seguridad, por lo que, además de disponer de un modelo de seguridad compuesto de características, es importante también identificar un conjunto de posibles amenazas que darán lugar a requisitos de seguridad (cuando sean consideradas) y por lo tanto a artefactos de análisis, diseño e implementación en los sistemas que se desarrollen.

En este sentido, MAGERIT define una taxonomía de amenazas clasificadas básicamente en desastres naturales, de origen industrial y errores y fallos no intencionados de los usuarios y del propio sistema.

F. Familia 27000

La familia de normas ISO/IEC 27000 está compuesta de un conjunto de documentos, todos ellos relacionados con la gestión de la seguridad. En concreto, la 27000 [21] incluye la definición de un vocabulario común sobre gestión de seguridad, la 27001 [22] proporciona un modelo para establecer, implementar, operar, controlar, revisar, mantener y revisar los sistemas de gestión de seguridad de la información, la

27002 [23] ofrece un código de buenas prácticas, la 27003 [24] unas guías de implantación, la 27004 [25] es relativa a métricas para la gestión de la seguridad, la 27005 [26] es sobre gestión de riesgos, la 27006 [27] muestra un cuerpo para la certificación de la seguridad y la 27007 [28] ofrece guías de auditoría. Esta familia de normas (todavía incompleta) representa un esfuerzo por la agrupación y unificación de estándares relativos a la gestión de la seguridad, y que se pretende que sea modelo de referencia en el futuro.

En su norma base, se define la seguridad de la información como la preservación de la confidencialidad, la integridad y la disponibilidad de la información. También considera, aunque en un menor nivel de importancia otras propiedades como la autenticación, la responsabilidad, el no repudio, y la fiabilidad. Todas estas características se encuentran rigurosamente definidas en dicho estándar.

G. COBIT versión 4.1

COBIT (Control Objectives for Information and Related Technology) [29] es un conjunto de buenas prácticas para el manejo de información creado por la Asociación para la Auditoría y Control de Sistemas de Información (ISACA). Este documento ayuda a entender los Sistemas de Información (o tecnologías de la información) y a decidir el nivel de seguridad y control que es necesario para proteger los activos de las compañías mediante el desarrollo de un modelo de administración de las tecnologías de la información. Uno de los aspectos más desarrollados en COBIT es la protección de la seguridad de la información, que aunque no es el único aspecto, aparece destacado frecuentemente. De hecho, por ejemplo, COBIT define un conjunto de ”criterios de información” necesarios para conseguir los objetivos de negocio, y la mayoría son relativos a seguridad, ellos son: eficacia, eficiencia, confidencialidad, integridad, disponibilidad, conformidad y fiabilidad. Todos estos conceptos se aproximan a la seguridad, aunque desde un nivel de abstracción muy alto y relativo al negocio y sus procesos.

COBIT proporciona, además, un conjunto de procesos que contribuyen al gobierno de las tecnologías de la información, y son agrupados en diversas categorías. Uno de estos procesos es dedicado monográficamente a la seguridad (se conoce como DS5 Asegurar Seguridad de Sistemas), y para el cual se definen un conjunto de objetivos de control como son gestión y planificación de TI, gestión de identidades y cuentas de usuario, pruebas, vigilancia y control de seguridad, definición de incidencias de seguridad, gestión de claves criptográficas, prevención, detección y corrección de software malicioso, seguridad en redes, intercambio de datos sensibles y protección de la seguridad de la tecnología. Estos objetivos de control representan aspectos que deben ser tenidos en cuenta para garantizar la seguridad de las TI, de modo que existirán probablemente requisitos relativos estos objetivos de control en los sistemas de información, y se tendrá que desarrollar funcionalidad que los resuelvan.

III. Estudio de las Características

En primer lugar, y como paso previo a la construcción del modelo de seguridad del software, hemos realizado un análisis de las distintas características, sub-características y sub-sub-características relacionadas con la seguridad, presente en los diversos modelos que hemos considerado en la sección anterior. Para ello, hemos construido la Tabla VI, donde se indican las distintas propiedades de seguridad de las propuestas analizadas (para mayor legibilidad, dichas propuestas han sido numeradas, y las correspondencias se ofrecen en la Tabla V), y se especifica en el cruce, la relación de cada una de esas propiedades con cada propuesta (en blanco cuando la propiedad no aparece en la propuesta, ”C” cuando aparece como característica, ”S” cuando aparece como sub-característica y ”U” cuando aparece identificado como parte de una sub-sub-característica).

Las diferencias en la consideración de unas propiedades y otras como características, sub-características, o incluso como parte de sub-sub-características, se justifica básicamente por dos hechos: el primero tiene que ver con el cambio en la consideración de la seguridad que ha habido en los últimos años (eso justifica la variación entre la ISO/IEC 9126 y la ISO/IEC 25010), mientras que el segundo es la orientación de las propuestas, ya que aquellas que consideran la seguridad como sub-característica, son propuestas que consideran la calidad de manera general, mientras que las que consideran estas propiedades como características, es porque se trata de propuestas claramente orientadas a la seguridad.

Tabla V
Enlace de columnas de la Tabla VI con los nombres de las propuestas

No. Prop.	Nombre Propuesta
1	ISO/IEC 9126 Modelo de Calidad Interna y Externa
2	ISO/IEC 9126 Modelo de Calidad en Uso
3	ISO/IEC 25010 Modelo de Calidad de Productos Software
4	ISO/IEC 25010 Modelo de Calidad en Uso
5	ISO/IEC 25010 Modelo de Calidad de Datos
6	Modelo de Firesmith
7	MAGERIT V2
8	Familia 27000
9	COBIT

Analizando las distintas propiedades de la Tabla VI, se puede observar que en muchas ocasiones aparecen propiedades que comparten similitud con otras, y que se diferencian en algún matiz acentuado por una propuesta en concreto. Por ejemplo, la propiedad Confidencialidad, aparece definida en prácticamente todas las propuestas, pero en cambio MAGERIT la define como Confidencialidad de Datos, aunque compartiendo en esencia la definición.

Por ello, se ha construido un grupo canónico de características que reduce el número inicial de propiedades de seguridad, uniendo aquellas propiedades relativas en esencia a aspectos muy cercanos, y dando lugar a propiedades de seguridad cuyo nombre probablemente ya está dentro del

Tabla VI
COMPARACIÓN DE PROPIEDADES DE SEGURIDAD

Característica \ Propuesta	1	2	3	4	5	6	7	8	9
Anonimato						U			
Auditoría de seguridad						S			
Autenticación						U		C	
Autenticidad de origen de datos							C		
Autenticidad de usuarios del servicio							C		
Autenticidad			S						
Autorización						U			
Confidencialidad	U		S		C	U		C	C
Confidencialidad de datos							C		
Conformidad			S						C
Conformidad (safety)				S					
Control de acceso						S			
Daño Comercial				S					
Daño medioambiental				S					
Detección de Ataques						S			
Disponibilidad	U				C		C	C	C
Fiabilidad								C	C
Identificación						U			
Integridad	U		S			S		C	C
Integridad de datos						U	C		
Integridad de hardware						U			
Integridad personal						U			
Integridad de software						U			
No Repudio			S			S		C	
Privacidad						S			
Protección física						S			
Responsabilidad			S					C	
Seguridad	S		C						
Seguridad (safety)		C		C					
Seguridad y salud del operador				S					
Seguridad y salud pública				S					
Trazabilidad de los datos							C		
Trazabilidad del servicio							C		

conjunto inicial de propiedades, pero cuya definición está enriquecida con los matices identificados en las propiedades iniciales de las que se parte (que en todo caso podrán actuar posteriormente como sub-características de ésta). Además, se han eliminado términos más generales que típicamente agregan algunos más detallados. En particular, se han suprimido los términos Seguridad y Seguridad con la orientación de Safety. También se ha eliminado la propiedad fiabilidad, ya que constituye una característica claramente diferenciada de seguridad, y así queda definida en el modelo de la ISO/IEC 25010.

Otro análisis de la Tabla VI indica que entre todas las propiedades de la seguridad, se puede observar una clara agrupación de aspectos relacionados con problemas de seguridad que son provocados intencionadamente, y por otro lado, en aspectos relacionados con problemas de seguridad fortuitos, que en principio pueden suceder sin que nadie los provoque intencionadamente (relacionados con el término inglés Safety). Todo esto redunda en otra categorización conforme al comentario anterior. Adicionalmente, en esta agrupación se considera la propiedad de conformidad, incluida en ambas categorías, ya que es importante desde ambos puntos de vista.

Estas clasificaciones de canonización e integración de características, así como de separación según la intencionalidad de los problemas de seguridad serán finalmente depuradas en la sección siguiente, con la propuesta de modelo de calidad de seguridad.

IV. MODELO PROPUESTO PARA LA SEGURIDAD DE PRODUCTOS SOFTWARE

Tras el análisis y depuración de propiedades de seguridad llevados a cabo en la sección anterior, se presenta el modelo de seguridad (Tabla VII), compuesto por dos grupos de características de seguridad. El primer grupo es relativo a propiedades de seguridad definidas para proteger al sistema ante ataques de seguridad, y el segundo grupo se refiere a propiedades de seguridad definidas para proteger al sistema de fallos y situaciones fortuitas. Cada característica define a su vez un conjunto de sub-características, que se refiere a algún matiz específico que aun estando relacionado con la característica a la que pertenece, tiene algún aspecto claramente diferenciador. Se puede comprobar que el modelo elaborado comparte cierta similitud con la ISO/IEC 25010, pero la enriquece ampliamente con aspectos identificados en propuestas más especializadas en seguridad.

Tabla VII
MODELO DE SEGURIDAD MEDUSAS

	Características	Sub-características
Protección ante ataques	Autenticidad	Autenticación
		Identificación
	Confidencialidad	Anonimato
		Privacidad
	Conformidad	
	Detección de ataques	
	Disponibilidad	
	Integridad	Integridad de datos
		Integridad de hardware
		Integridad personal
		Integridad de software
		Protección física
	No Repudio	
	Trazabilidad	
Protección ante accidentes		
	Conformidad (safety)	
	Daño Comercial	
	Daño medioambiental	
	Seguridad y Salud del operador	
	Seguridad y Salud pública	

El modelo considera las siguientes características y subcaracterísticas, relativas a la protección de los sistemas de información ante ataques provocados, cuyas definiciones son determinadas a partir de la extracción de conceptos integradores de los estándares y aproximaciones analizados en la Sección II.

- Autenticidad: Tiene que ver con el grado en el que se garantiza que los sujetos y recursos del sistema de información son auténticos.
  - Autenticación: Es relativo al grado en el que se verifica la identidad de los sujetos antes de interactuar con ellos.

  - Identidad: Es relativo al grado en que se identifican a los sujetos antes de interactuar con ellos.
- Confidencialidad: Es el grado en el que se asegura que la información es solamente accesible a sujetos autorizados.
  - Anonimato: Es el grado en que se impide el almacenamiento o descubrimiento de la identidad de los usuarios.
  - Privacidad: Es el grado en el que se asegura que la información de carácter personal, privado e íntimo es solamente accesible a sujetos autorizados.
- Conformidad: Es el grado en que los productos software se ajustan a los estándares, acuerdos, o regulaciones de leyes y otras recomendaciones similares de seguridad.
- Detección de ataques: Es el grado en que los intentos de ataque o los ataques realizados con éxito son detectados, almacenados y notificados.
- Disponibilidad: Es el grado en que se asegura que los sujetos autorizados tienen acceso a los datos y aplicaciones en el momento en que lo requieran.
- Integridad: Es el grado en que se protege a los componentes de los sistemas de información de alteraciones intencionada por parte de sujetos no autorizados.
  - Integridad de datos: Concepto de integridad aplicado a los datos.
  - Integridad del hardware: Concepto de integridad aplicado a los componentes hardware del sistema.
  - Integridad del personal: Es el grado en que se protege la seguridad de las personas ante posibles reacciones del sistema provocados intencionadamente.
  - Integridad del software: Es el grado en que se protege los componentes de software de corrupción intencionada.
  - Protección física: Es el grado en que el sistema se protege a sí mismo y a sus componentes de ataques físicos.
- No repudio: Es el grado en que se impide que una parte de una interacción pueda repudiar algún aspecto de la interacción.
- Trazabilidad: Es el grado en que se asegura que las acciones de un sujeto pueden ser trazadas inequívocamente y asociadas a dicho sujeto.

Con respecto al modelo de seguridad, para el caso de características de seguridad relativas a la protección de los sistemas de información ante accidentes no provocados, básicamente se heredan las propiedades definidas por la ISO/IEC 25010 en el modelo de Calidad en Uso.

V. Conclusiones

En el presente trabajo se han analizado los estándares y propuestas centradas en seguridad o con marcado enfoque en ella, donde se exponen un grupo de sus características inherentes y que han servido como base para el modelo de calidad propuesto. Dicho modelo es una propuesta integradora de conceptos con el fin de ofrecer una visión común en el área, tanto en lo que refiere a características y sub-características como a su definición formal.

Agradecimientos

Esta investigación es parte de los proyectos: MEDUSAS (IDI-20090557), financiado por el Centro para el Desarrollo Tecnológico Industrial, BUSINESS (PET2008-0136) concedido por el Ministerio de Ciencia e Innovación de España y SEGMENT (HITO-09-138) y SISTEMAS (PII2I09-0150-3135) financiados por la Consejería de Educación y Ciencia de la Junta de Comunidades de Castilla-La Mancha.

Referencias

[1] ITU. (2009). *ICT Security Standards Roadmap* Available: http://www.itu.int/ITU-T/studygroups/com17/ict/index.html
[2] S. Castano, et al.,*Database Security*: Addison-Wesley, 1995.
[3] ISO/IEC, "ISO/IEC 9126-1. Information Technology. Software Product Quality. Part 1: Quality Model.," ed, 1999.
[4] J. Swartz, "Security systems for a mobile world," *Technology in Society*, vol. 25, pp. 5-25, 2003.
[5] S. Posthumus and R. v. Solms, "A framework for the governance of information security," *Computers & Security*, vol. 22, pp. 638-646, 2004.
[6] J. Jürjens, "UMLsec: Extending UML for secure systems development," in *UML 2002 - The Unified Modeling Language, Model engineering, concepts and tools*, J. Jézéquel, et al., Eds., ed Dresden, Germany: Springer. LNCS 2460., 2002, pp. 412-425.
[7] J. Jürjens, *Secure Systems Development with UML*: Springer-Verlag, 2004.
[8] D. Basin, et al., "Model Driven Security: from UML Models to Access Control Infrastructures," *ACM Transactions on Software Engineering and Methodology*, vol. 15, pp. 39-91, January 2006.
[9] M. Hafner, et al., "SECTET: An Extensible Framework for the realization of Secure inter-organizational Workflows," *Internet Research*, vol. 16, pp. 491-506, 2006.
[10] E. Fernández-Medina and M. Piattini, "Designing Secure Databases," *Information and Software Technology*, vol. 47, pp. 463-477, 2005.
[11] C. Gutiérrez, et al., "Towards a Process for Web Services Security," *Journal of Research and Practice in Information Technology*, vol. 38, pp. 57-67, 2006.
[12] D. Mellado, et al., "A Common Criteria Based Security Requirements Engineering Process for Development of Securie Information Systems," *Computer Standards & Interfaces*, vol. 29, pp. 244-253, 2006.
[13] A. Rodríguez, et al., "Semi-Formal Transformation of Secure Business Processes into Analysis Class and Use Case Models: an MDA approach.," *Information and Software Technology*, 2010.
[14] ISO/IEC 25000, "Systems and software engineering - Software product Quality Requirements and Evaluation SQuaRE."
[15] D. Firesmith, "Specifying Reusable Security Requirements," *Journal of Object Technology*, vol. Vol. 3 (1), January-February., pp. 61-75, 2004.
[16] ISO/IEC, "ISO/IEC 15408. Common Criteria V3.," 2009.
[17] MAP, "Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información (MAGERIT - v 2)," 2005.
[18] CCN-CERT, "Últimos avances en ciberseguridad (9th NATO cyberdefense workshop). Revista auditoria y Seguridad (www.revista-ays.com). n23-junio: 70-71.," 2008.
[19] OECD, "The promotion of a culture of security for information systems and networks in OECD countries. DSTI/ICCP/REG(2005)1/FINAL, Organisation for Economic Co-operation and Development.," 2005.
[20] ISO/IEC, "ISO/IEC 13335 Information technology - Security techniques - Management of information and communications technology security.," 2004.
[21] ISO/IEC, "ISO/IEC 27000:2009. Information technology – Security techniques – Information security management systems – Overview and vocabulary," ed, 2009.
[22] ISO/IEC, "ISO/IEC 27001:2005. Information technology – Security techniques – Information security management systems – Requirements ", ed, 2005.
[23] ISO/IEC, "ISO/IEC 27002:2005. Information technology – Security techniques – Code of practice for information security management," ed, 2005.
[24] ISO/IEC, "ISO/IEC 27003:2010. Information technology – Security techniques – Information security management system implementation guidance," 2010.
[25] ISO/IEC, "ISO/IEC 27004:2009. Information technology – Security techniques – Information security management – Measurement," ed, 2009.
[26] ISO/IEC, "ISO/IEC 27005:2008. Information technology – Security techniques – Information security risk management," ed, 2008.
[27] ISO/IEC, "ISO/IEC 27006:2007. Information technology – Security techniques – Requirements for bodies providing audit and certification of information security management systems," 2007.
[28] ISO/IEC, "27007. Information technology – Security techniques – Guidelines for Information security management systems auditing.
[29] ITGI. (2007). COBIT 4.1: Marco de Trabajo, objetivos de control, directrices gerenciales y modelos de madurez. Available: www.itgi.org.

El Spyware como amenaza contra navegadores web

Sergio Castillo-Pérez*, José Alfredo Múrcia Andrés‡, Joaquin Garcia-Alfaro†‡

* Universitat Autònoma de Barcelona, Edifici Q, 08193, Bellaterra
† Institut Telecom, Telecom Bretagne, 35576, Cesson-Sevigne, France
‡ Universitat Oberta de Catalunya, Rambla de Poblenou 156, 08018, Barcelona

***Resumen*—En la última década se ha realizado un progreso sustancial en Internet y en las tecnologías basadas en el paradigma web. Aplicaciones relacionadas con educación, salud, banca, o incluso con relaciones entre individuos o grupos sociales, pueden beneficiarse mediante el uso de dichas tecnologías. Sin embargo, los ataques a sistemas pueden comprometer drásticamente la privacidad de los usuarios que hacen uso de las tecnologías web. En este contexto, la infección de sistemas mediante Spyware es un claro ejemplo. En este artículo analizamos la amenaza del Spyware como vía para comprometer la seguridad y privacidad de los recursos de los navegadores web.**

I. Introducción

El uso del paradigma web en todos los modelos de negocios y organizaciones está convirtiéndose en un aspecto omnipresente. De hecho, su uso aparece como una estrategia emergente en todos los tipos de aplicaciones software de las compañías [1]. Éste permite el diseño de aplicaciones totalmente interactivas que pueden potencialmente ser usadas por miles de usuarios alrededor del mundo. La existencia de nuevas tecnologías para mejorar las características del paradigma web tradicional permite a los ingenieros del software concebir nuevos servicios, que no están restringidos a un sistema operativo específico. Sistemas tradicionales de información relacionados con la educación, salud, banca o incluso sistemas de emergencia, pueden beneficiarse de esta tecnología.

La complejidad actual del paradigma web tiene, sin embargo, un impacto en la seguridad de los navegadores web y, de forma más precisa, en el tratamiento de sus recursos. Los ataques contra navegadores web pueden comprometer la seguridad y la privacidad de los usuarios. Esto puede tener serias consecuencias dada la omnipresencia de software malicioso, como el Spyware. El Spyware puede ser instalado de forma secreta en los navegadores web y robar datos sensibles, tales como identificaciones de usuarios, contraseñas o datos financieros [7]. Los navegadores web deben, por tanto, incluir mecanismos confiables para garantizar la seguridad y privacidad de sus usuarios. En este artículo damos una visión general de algunas técnicas usadas por el Spyware, y que pueden ser usadas por entidades maliciosas para violar la privacidad de los usuarios. Presentamos un escenario que muestra cómo la privacidad de un usuario accediendo a un servicio web puede ser violada por Spyware vinculado al navegador web. Seguidamente discutimos algunos mecanismos de defensa que pueden reducir el riesgo representado por la amenaza del Spyware.

Organización del artículo — La sección II presenta la amenaza del Spyware y desarrolla nuestro escenario de motivación. La sección III presenta una visión general sobre mecanismos de defensa para reducir el riesgo de la amenaza del Spyware. La sección IV concluye el artículo.

II. Spyware

El concepto Spyware (o software espía) es un término utilizado para catalogar al software malicioso (*Malware*) [11] que registra información de usuarios de forma no consentida, violando la privacidad de éstos. La información recolectada por este tipo de aplicaciones suele ser de distinta índole, tal como datos personales, números de tarjeta de crédito, hábitos de navegación web, contraseñas, pulsaciones de teclas, o captura de pantalla. Tal información es transmitida a terceras partes con finalidades como son el fraude electrónico [6], el marketing a través de publicidad web no consentida, u otras actividades normalmente maliciosas. Con la finalidad de conseguir su propósito, dicho software provoca diversos efectos en el comportamiento del sistema afectado, como la aparición de ventanas de navegación con publicidad no deseada, el secuestro del navegador web, instalación de puertas traseras (*backdoors*), modificación de los números de conexión a ISPs a otros con tarifas elevadas, etc. Asimismo, y con motivo de llevar a cabo su finalidad, la ejecución de estas aplicaciones suele conllevar la degradación del rendimiento del sistema, incrementando el uso de CPU, del espacio utilizado en disco, o del ancho de banda de red.

A lo largo del tiempo, el Spyware ha evolucionado incorporando sofisticados mecanismos propios de los rootkits. Estos mecanismos permiten al Spyware esconder su presencia a administradores o a software destinado a su detección. Así, estrategias como la ocultación de procesos, archivos o conexiones de red, o el uso de técnicas antidebugging o de criptografía, suelen ser características habituales en el Spyware de hoy en día. De forma análoga, y con el objetivo de no ser eliminados del sistema infectado, la inclusión de estrategias que dificultan su desinstalación son propiedades comunes en este tipo de software. Este conjunto de metodologías evasivas, junto a los mecanismos para la recopilación de información en si, suele conllevar la utilización de técnicas de programación que, en ocasiones, provocan cierta inestabilidad del sistema, dando lugar a un comportamiento inesperado de algunas aplicaciones.

En ocasiones, el concepto de Spyware es confundido con el de virus o gusano. A pesar de que ambos comparten una finalidad maliciosa, son radicalmente distintos en su modo de funcionamiento. Una de las diferencias primordiales que los distingue radica en el hecho de que el Spyware no suele auto-replicarse, es decir, un sistema afectado no propagará la infección a otros ordenadores. En el proceso de infección del Spyware podemos distinguir básicamente dos tipos de estrategia en función de si se requiere una interacción directa por parte del usuario, o si, por contra, la instalación se realiza de forma transparente.

En el primer caso, el Spyware suele venir camuflado en programas supuestamente legítimos y que, sin el consentimiento del usuario, es instalado junto a dicho software. Así, programas de tipo P2P, complementos para los navegadores web, software descargado de forma ilegal, o cracks, son ejemplos de programas que pueden esconder Spyware. En otras ocasiones, la vía de infección se basa en el acceso a una determinada web con componentes ActiveX o Applets Java especialmente preparados que, tras la autorización de su ejecución por parte del usuario, conllevan la instalación del Spyware. En cualquier caso, el proceso de infección requiere una aprobación de ejecución tácita por parte del usuario. La utilización de la ingeniería social suele estar presente en esta metodología, pudiéndose considerarse incluso un factor decisivo en el consecución del éxito para los atacantes. De esta manera, los atacantes utilizan estrategias para persuadir a los usuarios a descargar determinado software, o realizar determinadas acciones que conlleven la instalación del Spyware.

En el segundo caso, los mecanismos de infección del Spyware se basan en la explotación de vulnerabilidades en el software. De esta manera, la visita a una determinada web preparada haciendo uso de un navegador vulnerable, o la apertura de un archivo especialmente preparado mediante software que incluya deficiencias de seguridad, pueden provocar la ejecución de código arbitrario que conduzca a la instalación del Spyware. En este proceso de infección, la instalación del software espía suele pasar totalmente desapercibida para el usuario afectado, sin requerirse ningún tipo de acción a realizar que puede considerarse motivo de sospecha. Comúnmente, esta estrategia suele estar ligada a otros ataques previos como son la redirección mediante XSS o el DNS Spoofing entre otros. Una vez el proceso de infección ha tenido éxito, el software espía capturará la información de su interés, y la utilizará según la finalidad para la que fue programado. En base a los mecanismos que se usan para interceptar esta información, y de las estrategias utilizadas para evadir la detección y/o desinstalación, distinguimos entre Spyware en modo usuario y Spyware en modo de kernel.

Spyware en modo usuario — A esta categoría pertenece la mayor parte del Spyware que encontramos hoy en día, dado que su programación es más sencilla. Adicionalmente, dado que se ejecutan en modo de usuario, su detección y eliminación desentraña menos complejidad. Las estrategias empleadas en esta categoría para interceptar información, o los mecanismos para evadir su detección/desinstalación se basan principalmente en desviar el flujo de ejecución de las aplicaciones, cediendo el control a determinado código del Spyware. Para esto, existen básicamente dos técnicas. La primera se basa en modificar las direcciones de memoria donde son mapeadas las funciones de las librerías compartidas y utilizadas por las aplicaciones. En el segundo caso, la metodología se sustenta en aprovechar mecanismos de extensión que proporcionan determinados programas. Estos mecanismos permiten ceder el control a cierto código al producirse eventos asociados a dicha aplicación, permitiendo extender las funcionalidades del software de forma sencilla. Esta idea, que dota de flexibilidad a los programas, puede ser usada de forma maliciosa por parte del Spyware.

Spyware en modo de kernel — El Spyware en modo de kernel es conceptualmente igual que el Spyware en modo usuario, en el sentido que su funcionamiento se basa en desviar el flujo de ejecución de las aplicaciones. De nuevo, la ejecución es desviada a código del Spyware, el cual captura u oculta información en función de sus necesidades. La diferencia principal entre las dos categorías se basa en que en esta segunda el proceso de intercepción de información o evasión se realiza con un mayor nivel de privilegios. Concretamente, el código del software espía es cargado y ejecutado en el espacio de direcciones del núcleo del sistema operativo. Precisamente, esta característica de ejecución con privilegios de sistema les proporciona una mayor resistencia a ser detectados o eliminados, ya que las herramientas de detección o eliminación suelen ejecutarse en espacio de usuario. Esto implica una mayor complejidad en su código, lo que los hace menos comunes. La forma más habitual de desviar el flujo de ejecución por este tipo de Spyware se basa en modificar las tablas que contienen las direcciones asociadas a las llamadas al núcleo (*syscalls*).

II-A. Ejemplo de infección de un navegador web

Con el objetivo de ilustrar cómo la privacidad de un usuario puede ser vulnerada mediante Spyware, describimos en esta sección un hipotético escenario donde se pone de manifiesto este hecho. La técnica utilizada por el Spyware para interceptar la información se basará en explotar los mecanismos de extensión de ciertos programas — en nuestro caso, del navegador web. Esta característica está presente en la mayoría de los navegadores, y permite facilitar la adición de mejoras – comúnmente llamadas *extensiones* o *complementos* – por terceros programadores. Para conseguir esto, la rutina principal del navegador delega el flujo de ejecución a las nuevas funciones de las extensiones ante ciertos eventos, permitiéndoles la lectura y la modificación del DOM (*Document Object Model*). Así, por ejemplo, se puede ceder el control del navegador durante la carga de una web, la carga/descarga de archivos, etc. De la misma manera que las extensiones legítimas, esta tecnología puede ser explotada por el Spyware para introducir sus rutinas de captura de información.

Supongamos que un usuario malicioso desea obtener números de tarjetas de crédito válidos con el objetivo de perpetrar fraude electrónico. Asumamos también que dicho usuario malicioso descubre una vulnerabilidad asociada a una versión particular de un navegador web que permite la ejecución de código arbitrario. Analizando la vulnerabilidad, el usuario malintencionado podría preparar varios servidores web que contengan el código necesario para explotarla, así como forzaría a usuarios a visitar dichos servidores (e.g., mediante el uso de ingeniería social [6]). El acceso de las víctimas a estos servidores conllevaría la instalación del Spyware mediante la explotación de la vulnerabilidad. El objetivo del código malicioso instalado sería capturar y enviar hacia un sistema remoto los números de tarjeta de crédito introducidos por los usuarios durante sus transacciones electrónicas. En este contexto, el Spyware podría emplear los mecanismos de extensión del navegador con el propósito de conseguir su objetivo.

Para garantizar el éxito del escenario anterior, el atacante debe persuadir a un número elevado de potenciales víctimas (i.e., usuarios con un navegador vulnerable) a visitar los servidores preparados. A mayor número de víctimas persuadidas, mayor será el éxito del atacante. El factor tiempo también es un elemento decisivo en este proceso, dado que tan pronto como la vulnerabilidad sea reportada y corregida por los usuarios, menor será la probabilidad de éxito. Con la finalidad de incrementar la velocidad del proceso de infección, otras técnicas pueden ser utilizadas, tales como en envenenamiento de DNS, el envío de correo no solicitado (*spam*), o el cross-site scripting, redirigiendo a los usuarios a los servidores preparados.

Después de la infección, el Spyware — visto ahora como una extensión del navegador — permanece activo mientras analiza los datos que el navegador recibe o envía, a la espera de información de transacciones electrónicas (i.e., números de tarjetas de crédito, fechas de expiración, contraseñas, etc.). Toda la información capturada por el Spyware será enviada una vez post-procesada y protegida hacia un sistema remoto controlado por el atacante. Debemos notar que, a pesar de que el navegador puede usar criptografía para garantizar la autenticidad, integridad y confidencialidad de la información intercambiada con los servidores de comercio electrónico (e.g., usando el protocolo SSL/TLS), esto no protege la información robada por el Spyware.

Si analizamos el uso de SSL/TLS en el modelo de capas de TCP/IP, podemos observar que la información permanece protegida entre la capa de aplicación (HTTPS) y la capa de transporte (TCP). Sin embargo, dado que el Spyware que infectó el navegador es ejecutado en la capa de aplicación (i.e., HTTPS), éste intercepta la información antes de ser protegida por SSL/TLS. Asimismo, cabe destacar que el Spyware no provocará ningún mensaje sospechoso asociado a los certificados digitales de SSL/TLS, ya que el proceso de verificación no se ve afectado por la infección. De forma similar, el uso de códigos de verificación, tales como CVC2, CVV o CID, no ayudan tampoco a detectar o prevenir actividades maliciosas asociadas a un posible Spyware.

III. Técnicas de prevención

III-A. Mecanismos genéricos

Podemos considerar tres métodos principales para prevenir nuestros sistemas de las infecciones de Spyware. En primer lugar, fomentar las buenas prácticas por parte de los usuarios, manteniendo actualizado tanto el sistema operativo como las aplicaciones, impedir las descargas de software de fuentes no fiables o ignorar los correos y contenidos adjuntos de remitentes desconocidos. Desafortunadamente, estas prácticas no son cumplidas muchas veces por los usuarios, ni tampoco son completamente efectivas.

Otras formas de prevención, más técnicas, se basan en el diseño de patrones de protección. El objetivo de estos patrones es impedir la infección de un sistema o bien reducir el daño de la infección. Podemos agrupar dentro de esta categoría la utilización de anillos de protección. Dichos anillos establecen una estructura de confianza por capas en el sistema operativo y se complementan con hardware específico para poder realizar una separación efectiva entre procesos de confianza y procesos sospechosos. A través de esta solución, se pueden ofrecer distintos niveles de acceso a los recursos del sistema. De hecho, los anillos se organizan de forma jerárquica, estructurando aquellos dominios más privilegiados y confiables hasta los de menores privilegios y nivel de confiabilidad. De este modo, se reduce el riesgo de que procesos de tipo Spyware ataquen al núcleo del sistema operativo (lo que les permitiría obtener el control del sistema al completo [4]). Estos métodos han demostrado que, aunque reducen las consecuencias de una infección, no son totalmente efectivos.

El tercer mecanismo genérico, pensado especialmente para complementos o extensiones de software, consiste en verificar la autenticidad del código que se está ejecutando. Una primera solución consiste en la utilización de firmas digitales. Estas firmas se asociarán al código que ha de ser ejecutado y permiten verificar su autenticidad [17]. Para aquel código que no haya sido firmado, los usuarios deberán decidir entre: (1) no usar sus funcionalidades, o (2) exponerse a ciertos niveles de riesgo si dicho código es ejecutado. Con esta finalidad, se han dedicado esfuerzos a la investigación de mecanismos de tipo PCC (*Proof-Carrying Code*) y MCC (*Model-Carrying Code*). Ambas soluciones ofrecen una infraestructura para probar que el código se comportará de manera segura antes de su ejecución. Estas técnicas resultan eficientes en general a su aplicación al código móvil, pero ineficientes en el caso navegadores web [13].

III-B. Mecanismos específicos

Ante la ineficiencia de los métodos genéricos, las tendencias actuales se centran en la investigación de aplicaciones automáticas capaces de detectar y aislar código de tipo Spyware. Según la estrategia de análisis que utilicemos para la detección, podemos clasificar estos métodos en dos categorías principales: detección basada en análisis sintáctico (también conocida como detección de patrones o firmas) y detección basada en análisis semántico (detección por comportamiento).

Análisis Sintáctico — Consiste en la comparación del código a ejecutar contra una base de datos de firmas asociadas a código malicioso. El mayor inconveniente asociado a esta primera solución, al igual que ocurre con la totalidad de sistemas de detección basados en firmas, es la dependencia de una actualización periódica de la base de firmas. Por otro lado, la detección efectuada por este tipo de propuestas es fácil de evadir mediante técnicas de ofuscación de código [10], tales como el polimorfismo y el metamorfismo.

Análisis Semántico — Consiste en modelar la interacción de código binario con el sistema — a partir del cual se determina si se trata de software malicioso. Esta estrategia supera las deficiencias del análisis sintáctico, dado que las mutaciones no afectan al comportamiento final del Spyware. Por esto, esta estrategia es conocida también como detección basada en comportamiento. En función de cómo la información para el modelado es obtenida podemos encontrar dos categorías [5]: (1) análisis dinámico y (2) análisis estático.

Dentro de la primera categoría, encontramos, por ejemplo, el trabajo presentado por Vogt *et al.* en [12]. La propuesta consiste en un mecanismo de prevención de ataques XSS orientados al robo de información de usuario. Más concretamente, el mecanismo consiste en la utilización de un marcado de datos para la construcción y análisis de modelos dinámicos. Este mecanismo sufre, sin embargo, grandes deficiencias de rendimiento que lo hacen impracticable para la protección contra el Spyware en forma de extensiones de navegador. De hecho, el gran número de extensiones utilizadas actualmente en navegadores tales como Mozilla Firefox [13], hace que la solución propuesta por Vogt *et al.*, basada en la pre-evaluación e interpretación de cualquier código ejecutable que trate de interactuar con los recursos del navegador, resulte a la práctica muy ineficiente. Una mejora propuesta por Russo *et al.* en [14] trata de mejorar dicho rendimiento a partir de una reducción de las operaciones a monitorizar. En este caso, se propone la monitorización de tan sólo aquellas operaciones que interactúan con el DOM (*Document Object Model*) asociado a cada ejecución, redefiniendo su semántica. La propuesta no se centra únicamente en ataques de tipo Spyware mediante inyección de código malicioso, sino que más bien ofrece un mecanismo general para prevenir cualquier flujo considerado como de robo de información.

En [16], Moshchuk *et al.* proponen una estrategia de análisis basada en la utilización de dispositivos de tipo *proxy*. Estos dispositivos interceptan y analizan el contenido dirigido desde las aplicaciones web hacia los recursos del navegador. En el supuesto que dicho contenido no pueda ser analizado de manera estática, por motivos de rendimiento, por ejemplo, se propone realizar un renderizado a través de máquinas virtuales. Éstas se encargarán de supervisar los efectos de la ejecución del contenido interceptado, y determinarán si debe ser aceptado o rechazado. A diferencia de otros métodos que tratan de prevenir la intercepción de información confidencial, la propuesta se centra en evitar la instalación dentro del navegador de Malware procedente de la web analizada. Ciertas limitaciones relacionadas con el indeterminismo del paradigma web actual, ya que el flujo de ejecución puede variar en función del contexto y la interacción con los usuarios, podrían hacer impracticable este tipo de propuestas, debido al alto número de falsos positivos y negativos que se pueden llegar a generar.

Otras aproximaciones para la supervisión en tiempo de ejecución presentadas por Kirda *et al.* en [8], plantean un mecanismo de detección de Spyware aplicado específicamente a los componentes internos del navegador Internet Explorer (concretamente, las bibliotecas BHO — *Browser Helper Object* — asociadas al navegador). Esta solución propone una supervisión híbrida a partir de un análisis tanto dinámico como estático. Mediante el análisis dinámico, se supervisa la interacción de los componentes con el navegador, registrando las funciones COM (*Component Object Model*) que son invocadas como respuesta a los eventos de un usuario. Este registro permite identificar eventos específicos, que son posteriormente analizados más en detalle, a partir de un modelo estático. Este segundo análisis concluye con la creación de un grafo de flujos que caracteriza las reacciones asociadas a cada evento.

Nos gustaría destacar, por último, la propuesta presentada en [13] por Ter *et al.*. Esta propuesta ofrece protección de confidencialidad e integridad de los datos del usuario mediante la supervisión del acceso a datos y servicios realizados por extensiones de navegadores Mozilla Firefox a través de componentes XPCOM (*Cross Platform Component Object Model*) de Mozilla. La novedad de esta propuesta reside en la utilización de firmas del usuario asociadas a las extensiones del navegador. Estas firmas serán supervisadas tanto en el proceso de instalación de las extensiones, como durante la carga y ejecución de las extensiones. El mayor inconveniente relacionado con la propuesta, al igual que sucede con los mecanismo de prevención genéricos tratados en la sección III-A, es la dependencia del buen comportamiento por parte de los usuarios, lo que en general comporta el fracaso de la medida de prevención.

IV. Conclusiones

En este artículo se ha presentado un análisis de soluciones existentes contra la infección de sistemas mediante código malicioso (Malware) de tipo Spyware. El concepto de Spyware (o software espía) ha sido presentado como una evolución de Malware tradicional, cuyo objetivo es la interceptación no consentida de información perteneciente a los usuarios de los recursos de un navegador web. Dicha interceptación pretende violar, por lo tanto, la privacidad de usuarios de aplicaciones web relacionadas con educación, salud, banca, o redes sociales. Se han analizado a continuación algunas de las técnicas que pueden ser utilizadas por código Spyware para infectar los recursos de un sistema. Finalmente, se ha concluido el análisis con un resumen de soluciones generales para la prevención de este tipo de infecciones. Una presentación más elaborada de nuestro estudio así como los resultados iniciales de una propuesta propia serán tratados en un informe futuro.

Agradecimientos — Este trabajo ha sido financiado por el Ministerio de Ciencia y Educación, a través de los proyectos TSI2007-65406-C03-03 E-AEGIS y CONSOLIDER-INGENIO CSD2007-00004 ARES.

REFERENCIAS

[1] Cary, C., Wen, H. J., Mahatanankoon, P. (2004). A viable solution to enterprise development and systems integration: a case study of web services implementation. International Journal of Management and Enterprise Development, 1(2):164–175, Inderscience.

[2] Christodorescu, M., Jha, S., Seshia, S. A., Song, D., Bryant, R. E. (2005). Semantics-Aware Malware Detection. IEEE Symposium on Security and Privacy (S&P'05), pp.32-46.

[3] Egele M., Kruegel C., Kirda, E., Yin, H., Song, H. (2007). Dynamic Spyware Analysis. *USENIX Annual Technical Conference*.

[4] Embleton, S., Sparks, S., Zou, C. (2008). SMM Rootkits: a New Breed of OS Independent Malware. In *SecureComm '08: Proceedings of the 4th international conference on Security and privacy in communication netowrks*, pp. 1–12, ACM.

[5] Jacob, G., Debar, H., Filiol, E. (2008). Behavioral Detection of Malware: From a Survey Towards an Established Taxonomy. *Journal in Computer Virology*, 4(3):251–266.

[6] Garcia-Alfaro, J., Cuppens, F., Autrel, F., Castella-Roca, J., Borrell, J., Navarro, G., Ortega-Ruiz, J. (2005). Protecting On-line Casinos against Fraudulent Player Drop-out. IEEE International Conference on Information Technology. IEEE Computer Society.

[7] Hu, Q., Dinev, T. (2005). Is Spyware an Internet Nuisance or Public Menace?. *Communications of the ACM*, SPECIAL ISSUE: Spyware, 48(8):61-66.

[8] Kirda, E., Kruegel, C., Banks, G., Vigna, G., Kemmerer, R. A. (2006). Behavior-Based Spyware Detection. *USENIX Security '06*, Vancouver, Canada.

[9] Lee, Y., Kozar, K.A. (2005). Investigating Factors Affecting the Adoption of Anti-Spyware Systems.. *Communications of the ACM*, SPECIAL ISSUE: Spyware, 48(8):72-77.

[10] Moser, A., Kruegel C., Kirda E. (2007). Limits of Static Analysis for Malware Detection. *Computer Security Applications Conference*, ACSAC 2007, pp. 421-430.

[11] Skoudis, E., Zeltser, L. (2004). Malware: Fighting Malicious Code. *Prentice Hall PTR*.

[12] Vogt, P., Nentwich, F., Jovanovic, N., Kirda, E., Kruegel, C., Vigna, G. (2007). Cross-Site Scripting Prevention with Dynamic Data Tainting and Static Analysis.. *Proceeding of the Network and Distributed System Security Symposium (NDSS?07)*.

[13] Ter Louw M., Soon Lim, J., Venkatakrishnan, V.N. (2007). Extebsible Web Browser Security. *Lecture Notes in Computer Science*. Springer.

[14] Russo, A., Sabelfeld, A., Chudnov, A. (2009). Tracking Information Flow in Dynamic Tree Structures. *M. Backes an P. Ning (Eds.): ESORICS 2009*, LNCS pp. 86-103, Springer.

[15] Hallaraker, O., Vigna, G. (2005). Detecting Malicious JavaScript Code in Mozilla. *10th IEEE International Conference on Engineering of Complex Computer Systems*, pp 85-94.

[16] Moshchuk, A., Bragin, T., Deville, D., Gribble, S.D., Levy H.M. (2007). SpyProxy: Execution-based Detection of Malicious Web Content. *16h USENIX Security Symposium*.

[17] R. Sekar, V.N. Venkatakrishnan, Samik Basu, Sandeep Bhatkar, Daniel C. Du Varney. (2003). Model-Carrying Code: A Practical Approach for Safe Execution of Untrusted Applications.. *19th ACM symposium on Operating systems principles*, ACM.

Patrones de Seguridad: ¿Homogéneos, validados y útiles?

Santiago Moral-García
Grupo Kybele
Dep. de Lenguajes y Sistemas Informáticos II
Universidad Rey Juan Carlos
Email: santiago.moral@urjc.es

Roberto Ortiz
S21sec Labs. SOC
Grupo S21sec Gestión S.A.
Email: rortiz@s21sec.com

Belén Vela
Grupo Kybele
Dep. de Lenguajes y Sistemas Informáticos II
Universidad Rey Juan Carlos
Email: belen.vela@urjc.es

Javier Garzás
Grupo Kybele
Dep. de Lenguajes y Sistemas Informáticos II
Universidad Rey Juan Carlos
Kybele Consulting
Email: javier.garzas@urjc.es; javier.garzas@kybeleconsulting.com

Eduardo Fernández-Medina
Grupo GSyA. Dep. de Tecnologías
y Sistemas de Información
Universidad de Castilla-La Mancha
Email: Eduardo.FdezMedina@uclm.es

***Resumen*—Actualmente, la seguridad de la información es uno de los pilares principales en la gestión de las organizaciones. La evolución de los sistemas conlleva un aumento de su complejidad, y esto a su vez ha derivado en un incremento de los ataques a los sistemas de información, ya que hay muchas más posibilidades de que los atacantes encuentren nuevas vulnerabilidades. Por todo esto, es necesario proveer a los diseñadores de sistemas de soluciones fiables para minimizar este número de ataques y conseguir un menor impacto en su organización. Los patrones de seguridad son un buen mecanismo para aportar soluciones a problemas concretos de seguridad, ya que proporcionan soluciones estructuradas que solventan problemas recurrentes. Existen muchas propuestas enfocadas a la creación de nuevos patrones de seguridad, pero en la actualidad no se utilizan unas pautas homogéneas para realizar su descripción. Además, la mayoría de patrones existentes difícilmente pueden ser aplicables en el diseño de sistemas complejos, ya que en su descripción no contemplan la complejidad de las instalaciones reales. En este artículo se va a realizar una síntesis de un conjunto de propuestas que describen patrones de seguridad, basada en una revisión sistemática realizada anteriormente. Posteriormente se va a realizar un análisis en relación al contexto en el que se utilizan los patrones de seguridad, las plantillas y elementos que se usan a la hora de describir este tipo de patrones y además se estudiará la aplicabilidad de éstos en entornos reales. Finalmente se realizará una discusión para detectar las carencias ofreciendo a su vez una serie de propuestas de mejora.**

***Palabras Clave*—patrones, patrones de seguridad, seguridad de la información.**

I. Introducción

En los últimos años los avances tecnológicos están mejorando multitud de aspectos relacionados con el diseño y desarrollo de los Sistemas de Información (SI), provocando un crecimiento de la funcionalidad y aplicación de la que son dotados estos sistemas. Este crecimiento conlleva un aumento de la complejidad de los SI, incrementando el impacto de los ataques informáticos, ya que los atacantes tienen más posibilidades de encontrar nuevas vulnerabilidades en los sistemas, tales como Cross-Site Scripting, inyección de código, ejecución de ficheros maliciosos, etc. [1].

Por esta razón, la seguridad de la información es una de las principales preocupaciones que tienen las organizaciones en los últimos años. Por un lado, las compañías quieren evitar que su información esté en peligro y por otro lado, hay un incremento de ataques informáticos debido a los beneficios que pueden obtener los atacantes con la información que sustraen de las organizaciones. Debido a estos ataques, los diseñadores de SI deben incluir requisitos de seguridad a la hora de diseñar sus sistemas, es decir, deben asegurar la confidencialidad, integridad y disponibilidad de los datos siempre que sea necesario, para así, proteger los activos de información de la organización. La importancia en el diseño de sistemas seguros ha aumentado desde que la mayoría de los ataques a sistemas de software están basados en vulnerabilidades causadas por un deficiente diseño y desarrollo de las funcionalidades de las que se dota a los sistemas [2]. Para evitar estas deficiencias, los diseñadores de SI necesitan elaborar soluciones específicas para resolver problemas relacionados con las vulnerabilidades de seguridad, y así minimizar el número de ataques exitosos contra sus SI.

Los patrones son una buena forma de satisfacer esta necesidad, ya que describen un problema que ocurre una y otra vez en un entorno, proporcionando una solución documentada y validada que puede ser usada múltiples veces [3]. Una de las principales ventajas de los patrones es que combinan experiencia y buenas prácticas en el diseño de SI [4], haciéndolo más eficiente. También es importante resaltar que los patrones no son una solución a un problema propuesto, sino una guía homogénea que documenta cómo problemas similares fueron resueltos anteriormente.

Por lo tanto, los diseñadores de SI podrían usar patrones de seguridad para obtener soluciones fiables relacionadas en este campo, ya que son un buen mecanismo para optimizar

el proceso de decisión a la hora de resolver un problema de seguridad recurrente. Otra ventaja que encontramos en los patrones de seguridad es que incorporan un conocimiento extenso acumulado sobre seguridad de forma estructurada, proporcionando una serie de pautas para el diseño, construcción y evaluación de SI seguros [5].

La utilización de patrones de seguridad como guía para diseñar un SI seguro es una práctica bastante extendida [6, 7, 8, 9]. De hecho, en los últimos años, el número de patrones de seguridad publicados ha crecido de manera considerable [10, 11, 12, 13, 14]. Sin embargo, existe una gran variedad y diversidad en las pautas de descripción de cada una de las propuestas [15, 16, 17, 18], incluso, en repetidas ocasiones, se han propuesto varios patrones diferentes que dan respuesta al mismo conjunto de requisitos o problemas de seguridad [19, 20], es decir, existe una superposición de soluciones.

En este artículo se va a realizar una síntesis sobre las principales propuestas que describen patrones de seguridad, basada en una revisión sistemática que se ha realizado previamente, siguiendo la propuesta de [21]. El objetivo principal del trabajo que aquí se presenta es realizar un análisis sobre el contexto en el que se emplean los patrones de seguridad, las plantillas utilizadas para describirlos y los elementos usados en estas descripciones. Adicionalmente, se va a comprobar cuántas propuestas están basadas y validadas en casos reales y cuántas en ejemplos teóricos. Este estudio ayudará a verificar la aplicabilidad que tienen los patrones de seguridad analizados en procesos de diseño de SI reales.

El resto del artículo se organiza de la siguiente manera. En la Sección II se realiza una síntesis de un conjunto de propuestas que describen patrones de seguridad. La Sección III muestra los resultados obtenidos y expone una discusión sobre los mismos. Finalmente, la Sección IV presenta las principales conclusiones y los trabajos futuros.

II. Síntesis de Propuestas de Patrones de Seguridad

En esta sección se va a realizar una síntesis de las principales propuestas que describen patrones de seguridad. Debido a la diversidad de soluciones que se proponen, agruparemos los trabajos según la problemática que solucionan los patrones que describen.

A. Patrones de seguridad para comunicaciones seguras

En este apartado se han agrupado los artículos relacionados con soluciones de seguridad centradas en el ámbito de las comunicaciones entre los distintos sistemas, y el envío y recepción de mensajes que realizan.

En [22] se presentan cuatro patrones de seguridad que pueden ser utilizados para el diseño seguro de sistemas VoIP, ya que proponen soluciones que pueden controlar muchos de los posibles ataques brindando un entorno de trabajo para ayudar a los diseñadores a aplicar la seguridad en sus SI.

En [23] se proponen tres patrones de diseño para las implementaciones de sistemas VoIP en relación con problemas de seguridad específicos. Se propone una técnica de cifrado y descifrado para paquetes de voz y añaden una nueva propuesta de generación de claves. También desarrollan un módulo de IPsec para sistemas VoIP en entornos Cliente/Servidor.

En [24] se presenta un patrón para reforzar el canal de comunicaciones entre diferentes sistemas. Este patrón puede ayudar a los diseñadores de SI agregando controles de seguridad en la fase de procesado del flujo de datos.

B. Patrones de seguridad para control de acceso e identificación seguros

En este apartado se incluyen trabajos que muestran patrones para aumentar la seguridad de los SI, reforzándolos con mecanismos efectivos de identificación, autorización, autenticación y control de acceso.

En [25] se propone un lenguaje de patrones para el sistema de gestión de identidades, compuesto por tres patrones. Este lenguaje de patrones está basado en SAML (Security Assertion Markup Language), que proporciona un formato específico para la comunicación de información acerca de la identidad de los diferentes dominios de seguridad.

En [26] se describe una solución de control de acceso, para los datos generados por sensores inalámbricos. Esta propuesta está formada por tres patrones de seguridad que definen un modelo abstracto de control de acceso basado en criptografía.

En [27] se propone un patrón de seguridad que describe el uso de la identificación de la información de credenciales para definir la autenticación y el control de acceso. Este patrón está descrito para ser usado en sistemas distribuidos con el fin de asegurar estos requisitos.

En [28] se describen varios patrones para abordar aspectos relativos a las sesiones en modelos de control de acceso. Los autores muestran un patrón que controla el acceso de las diferentes sesiones, describiendo cómo una sesión puede limitar el derecho de un usuario. Además, se utilizan dos patrones más, que combinados con el patrón anterior, pueden constituir un patrón específico de control de acceso. Por último, esta propuesta muestra un entorno de trabajo real basado en el conjunto de patrones descrito.

C. Patrones de seguridad para garantizar la privacidad

En este apartado se incluyen los trabajos que proponen soluciones al problema de la privacidad tratando de preservar este aspecto utilizando patrones de seguridad. La privacidad es muy relevante en los intercambios de datos personales entre los usuarios y sistemas.

En [29] se presenta un conjunto de patrones para la estandarización del desarrollo de políticas de privacidad con el fin de ser utilizado en sitios web. Estos patrones consideran principalmente aspectos relacionados con la seguridad, la información del usuario y la privacidad. Además, los autores muestran un ejemplo ficticio de una política de privacidad en la que se combinan todos los patrones descritos en la propuesta, pudiendo ser aplicada en un sitio web.

En [18] se presentan dos patrones de seguridad: uno para la manipulación de cookies, que protege la identidad de los

usuarios cuando tienen acceso a un sitio web y otro, que permite a los usuarios utilizar un servicio de correo electrónico sin revelar su propia identidad.

En [30] los autores enfocan su trabajo en mejorar los patrones de privacidad existentes. Además, para reforzar este escenario se describen tres patrones adicionales. Estos nuevos patrones pueden ayudar a preservar la privacidad pretendiendo que las organizaciones online, los diseñadores de páginas web y los usuarios puedan utilizar información personal sin ningún problema de seguridad.

D. Patrones de ataque y de mal uso

En este apartado se incluyen los trabajos que describen otro tipo de patrones de seguridad: los patrones de ataque y los patrones de mal uso. En este tipo de patrones los autores se colocan en el lado del atacante y describen paso a paso todos los elementos del ataque a un sistema. Para ello definen el contexto del ataque, exponen los patrones de seguridad que lo neutralizan y proponen mecanismos para trazar las evidencias que deja el ataque una vez que ha ocurrido.

En [16] se propone un patrón de uso indebido y se presenta un modelo que expone la estructura de este tipo de patrón. De manera similar al anterior trabajo, en [31] se presenta un patrón de ataque, que proporciona una descripción específica de los objetivos del ataque. Además, se presenta como ejemplo ficticio un ataque de denegación de servicio en las redes de tipo VoIP para demostrar la efectividad del patrón expuesto.

E. Patrones de seguridad para mejorar la relación de confianza

En este apartado se incluyen los trabajos que proponen patrones de seguridad para reforzar las relaciones de confianza entre el usuario y los sistemas o entre dos usuarios, para tratar de cumplir los requisitos fundamentales de seguridad.

En [32] se presenta un patrón de seguridad para desarrollar una interfaz gráfica de usuario segura. Este patrón puede ayudar a reforzar los sistemas de interfaz gráfica de usuario y evaluar su uso en diferentes ámbitos. Además, se muestra cómo analizar los requisitos de seguridad para fomentar la confianza, preservando al mismo tiempo la flexibilidad que demandan las interfaces gráficas de usuario.

En [33] se describen patrones para reforzar las relaciones de confianza entre diferentes usuarios. Estos patrones permiten que dos usuarios puedan verificar mutuamente el perfil del otro sin revelar su identidad.

F. Otros patrones de seguridad para construir sistemas seguros

En este apartado se muestran varias propuestas para construir sistemas seguros utilizando patrones de seguridad, tanto de diseño como arquitectónicos.

En [34] se propone un patrón para reforzar las arquitecturas basadas en tres capas. Este patrón puede ser aplicado a sistemas distribuidos y enfocado a la ejecución de aplicaciones complejas y heterogéneas. Hay distintos debates sobre las propiedades del patrón arquitectónico en tres capas, así como varios patrones desarrollados [35, 36], pero ninguno de éstos considera la seguridad.

En [37] se describen patrones de seguridad para la representación de los procesos y subprocesos de los sistemas operativos. Como los sistemas operativos son muy críticos, los autores proponen varios patrones para resolver sus problemas de seguridad.

En [38] se introducen patrones para monitorizar las propiedades de seguridad básicas de un SI. Con ellos se puede comprobar, en tiempo de ejecución, la robustez de los requisitos generales de seguridad de un SI.

III. Resultados y Discusión

Como se ha mostrado en la sección anterior, existe una gran variedad de propuestas que trabajan descubriendo nuevos patrones de seguridad en relación a las necesidades de los SI. En esta sección se van a analizar, por un lado, los criterios de descripción utilizados en las propuestas sintetizadas y por otro lado, los entornos en los que son aplicados los patrones descritos. Finalmente, se mostrará una discusión en relación a los resultados obtenidos y se propondrán una serie de mejoras.

En la Figura 1 se muestran horizontalmente las referencias de los trabajos sintetizados y el contexto al que pertenece cada uno de ellos, siguiendo la estructura de los apartados descritos en la sección anterior. Verticalmente se muestran las plantillas que han sido utilizadas para describir las propuestas, detallando los elementos utilizados en la descripción de cada uno de los patrones de seguridad. Las plantillas de descripción incluidas en la Figura 1 son, la plantilla resultante de la fusión de elementos de la plantilla propuesta por Gang of Four [39] adaptada a los patrones de seguridad y de la plantilla propuesta por Buschmann et al. [40] denominada PoSA, la plantilla propuesta en el proyecto SERENITY utilizada en [26], la plantilla propuesta por Alexander [3], y la plantilla de descripción de patrones en forma de eventos de cálculo [38]. Se sombrearán las casillas que correspondan a las plantillas o elementos de descripción que utilicen cada una de las propuestas respectivamente. Como se puede extraer de la Figura 1 y más concretamente de las columnas que representan las plantillas utilizadas en la descripción de patrones, no existe una plantilla estándar que sirva de guía para la descripción de patrones de seguridad que pueda ser utilizada por los expertos en esta materia. Esta situación provoca una variabilidad significativa en relación a los elementos que componen un patrón de seguridad. Como se observa en las columnas que se refieren a los elementos utilizados en las descripciones de patrones, cada autor describe los patrones siguiendo sus propias directrices, aunque existen algunos elementos comunes de las diferentes plantillas [18, 26, 30]. Incluso utilizando la misma plantilla, algunos autores no repiten en su totalidad todos los elementos de ésta, probablemente fruto de evolución en sus propuestas y tratando de mejorar la usabilidad de los patrones, optan por añadir nuevos elementos [25, 27, 29].

Los elementos más utilizados en las descripciones de patrones de seguridad son la tripleta ”Contexto”, ”Problema” y ”Solución” propuesta en [3]. Esto demuestra que existe

		Plantillas de Representación				Elementos para describir los patrones																								
Contexto de Patrones	**Referencias**	Gang of Four + PoSA	SERENITY	Alexander	Eventos de Cálculo	Nombre	Intención	Ejemplos	Contexto	Problema	Solución	Implementación	Consecuencias	Patrones Relacionados	Usos conocidos	Casos Reales/Escenarios	Estructura	Dinámica	Ejemplos resueltos	Ver también	Precondiciones	Propiedades	Características	Fuerzas	*	Reglas	Variantes	Evidencias
Comunicaciones	[22]	■							■	■	■	■	■	■	■														
	[23]	■							■	■	■	■	■																
	[24]	■						■	■	■	■	■	■		■		■	■	■	■									
Gestión de la Identidad	[25]	■							■	■	■	■	■	■	■		■	■											
	[26]		■						2	2	1		■			■					■	■	■						
	[27]	■						■	■	■	■	■	■	■	■		■	■	■										
	[28]	■						■	■	■	■	■	■	■	■				■										
Privacidad	[29]	■							■	■	■		■	■	■									■	■				
	[18]			■		■			■	■	■			■										■					
	[30]	■				■			■	■	■		■		■														
Punto de Vista del Atacante	[16]	■				■	■		■	■	■		■	■	■														
	[31]	■				■	■		■	■	■		■																■
Relaciones de Confianza	[32]	■					■	■	■	■	■		■	■	■		■	■	■										
	[33]		■						2	2	1		■			■					■	■	■						
Arquitecturas 3 capas	[34]	■							■	■	■	■					■	■											
Sistemas Operativos	[37]	■						■	■	■	■	■	■	■	■		■	■	■								■		
Monitorización de la Seguridad	[38]				■			■																		■			

1. Estos trabajos desarrollan el elemento de descripción "Solución" con más detalle
2. Estos trabajos contienen una sección con los siguientes elementos: problema/requisitos y contexto
*. Sección que contiene los siguientes elementos: Seguridad / Acceso / Elección / Transparencia

Fig. 1. Características de las propuestas estudiadas

una necesidad por parte de los investigadores de definir una serie de conceptos básicos a la hora de documentar un patrón descubierto, pero al carecer de una plantilla estándar para exponer los patrones de seguridad, se genera una diversidad muy destacada en las distintas descripciones. Debido a esta diversidad aumenta la complejidad para realizar un catálogo homogéneo de patrones de seguridad, ya que es difícil unificar toda la literatura existente sobre éstos. Este hecho también puede provocar que los diseñadores de SI tengan cada vez más dificultad a la hora de seleccionar los patrones más apropiados para unos determinados requisitos de seguridad dados [41]. Localizado este problema se detecta la necesidad de diseñar un conjunto de pautas que recojan una serie de características esenciales para la descripción de los patrones de seguridad, con el fin de conseguir un catálogo homogéneo. La principal aportación de este catálogo sería conseguir soluciones equivalentes entre distintos diseñadores de SI seguros.

Adicionalmente al análisis realizado sobre las distintas descripciones utilizadas en el diseño de patrones de seguridad, se ha realizado un análisis sobre los entornos en los que son validadas las propuestas seleccionadas. En la Figura 2 se muestran los porcentajes de las propuestas que han sido validadas en entornos reales (SI), las que han sido validadas parcialmente en entornos reales (PARCIALMENTE), es decir, han simulado un entorno real a pequeña escala y las que están basadas en casos de laboratorio (NO). Como se puede observar en la Figura 2, el 59% de las propuestas seleccionadas están basadas en ejemplos teóricos o casos de laboratorio, el 29% de ellas están validadas en un entorno real simulado, mientras que sólo el 12% de las propuestas han sido validadas completamente en entornos reales.

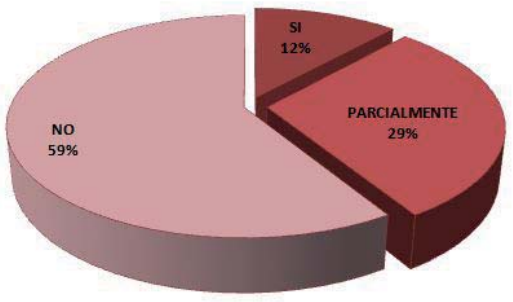


Fig. 2. Propuestas validadas en entornos reales

Tras realizar este análisis se detecta una carencia de visión real en los patrones de seguridad, es decir, no se expone una aproximación práctica específica para el diseño de SI seguros basado en patrones de seguridad. Este aspecto difiere en su totalidad de la propia definición de patrón, que

precisamente declara que una de las principales bondades de los patrones es que proporcionan soluciones validadas que resuelven problemas similares. Este hecho puede provocar una escasa aplicabilidad de los patrones de seguridad en diseños de SI reales, ya que probablemente éstos no han sido generados cómo conclusión de la solución a un problema en un entorno real complejo.

Un patrón de seguridad debería servir para simplificar la toma de decisiones de un ingeniero de seguridad de la información a la hora de diseñar un nuevo SI o implantar un nuevo sistema dentro de un sistema mayor, reduciendo el tiempo y el coste del análisis de seguridad. Desde nuestra experiencia, a la hora de realizar el análisis de seguridad de un SI en un entorno real es necesario considerar los aspectos relativos a: a) el número de elementos físicos y lógicos que componen el SI; b) la gestión y aprovisionamiento de usuarios; c) el proceso de copias de seguridad y sobre qué elementos habría que realizarlas; d) la trazabilidad de la solución aportada; e) la expansión de la solución de forma masiva; f) el impacto en los parámetros básicos de un SI, la memoria, la capacidad de proceso, el almacenamiento, el ancho de banda consumido, etc. Por todo esto, los patrones de seguridad basados en casos de laboratorio difícilmente pueden ser utilizados en un proceso de diseño de un SI real, ya que la mayoría no tienen en cuenta la complejidad de las instalaciones reales incumpliendo las premisas anteriores cuando son diseñados. En caso de ser utilizados por un ingeniero de SI, existe la posibilidad que aumente el tiempo y el coste del análisis de seguridad en el ciclo de vida del diseño del SI.

Para concluir este apartado se van a exponer las necesidades detectadas tras el análisis realizado. En primer lugar, hay que destacar la complejidad que presenta ofrecer actualmente, tanto al experto como al no experto en seguridad, una guía de soluciones reutilizables, a fin de que sea usada para diseñar un sistema seguro, ya que como queda demostrado, no es tarea fácil alinear los diferentes criterios de descripción en el ámbito de los patrones de seguridad. Por este motivo, se detecta la necesidad de establecer una metodología dentro del ámbito de la Seguridad de la Información en la que paso a paso, se describa cómo resolver un problema utilizando patrones de seguridad. Esta metodología debería aportar soluciones equivalentes entre distintos diseñadores de SI, con el fin de que esas soluciones puedan ser utilizadas por cualquiera que lo necesite, beneficiándose sin necesidad de tener conocimientos avanzados en el campo de la seguridad. Las soluciones aportadas llegarían a ser reutilizables y exportables, ya que recogerían todas las características técnicas del sistema, las personas involucradas en la solución planteada, etc. Además, se detecta una clara necesidad de crear soluciones de seguridad estructuradas en forma de patrones que reflejen soluciones validadas en entornos reales complejos siguiendo las premisas que se han expuesto anteriormente. Finalmente, se detecta la necesidad de enriquecer y completar la descripción de los patrones de seguridad actuales, con un conjunto de elementos que describan los aspectos principales para un diseñador de SI a la hora de implementar la solución en instalaciones reales, con el fin de aumentar la aplicabilidad de estos patrones ya descubiertos en este tipo de entornos.

IV. Conclusiones y Trabajos Futuros

En este trabajo se han sintetizado un conjunto de propuestas que describen patrones de seguridad. Se puede observar que el número de estudios dentro de este ámbito es muy elevado y abarca distintos contextos, encontrando gran heterogeneidad en el proceso de creación de patrones descrito por los distintos investigadores. Cada uno de los enfoques selecciona un conjunto distinto de elementos para realizar la descripción de los patrones, provocando un aumento de complejidad al realizar una clasificación homogénea de los patrones existentes. Por este motivo, los diseñadores de los SI pueden tener una mayor dificultad a la hora de seleccionar una serie de patrones apropiados para diseñar sus sistemas seguros. Por todo esto, se cree necesario definir un conjunto de pautas de descripción de patrones de seguridad que sea aceptado y utilizado por todos los investigadores relacionados con este campo.

Partiendo de la base de que los patrones por definición son un mecanismo validado, en la realización del estudio se han encontrado muy pocas propuestas validadas en entornos reales complejos. Por nuestra experiencia, los patrones de seguridad deberían considerar aspectos tales como medidas volumétricas de los parámetros básicos de un SI (memoria, capacidad de proceso, almacenamiento, etc.), gestión de usuarios, medidas de complejidad de uso, tanto para administradores como para usuarios finales, etc. En la actualidad, los patrones existentes no contemplan estos aspectos, por lo que se propone una profunda investigación para descubrir nuevos patrones que sí los reflejen. Además, se considera necesaria una evolución de los patrones existentes para cubrir las necesidades anteriores. Por último, se propone el desarrollo de una metodología de seguridad basada en patrones que guie al usuario a la hora de afrontar un problema en este ámbito. Esta metodología debería ser útil para cualquier tipo de diseñador de sistemas de seguridad, ya sea experto o no en este ámbito.

En trabajos futuros, se pretende abordar el desarrollo de una serie de pautas que recojan un conjunto de características principales para la definición de nuevos patrones de seguridad, con el fin de mantener un criterio equivalente entre las distintas propuestas que se vayan realizando. También, se pretende descubrir nuevos patrones de seguridad que cumplan los requisitos expuestos para aplicarlos en entornos reales. Finalmente, se propondrá una metodología de seguridad basada en patrones que aporte soluciones homólogas, validadas y reutilizables. Esta metodología servirá para dar soporte a los diseñadores de SI seguros para que paso a paso sepan cómo afrontar un problema de seguridad guiándoles para que lo resuelvan de la manera más óptima y eficiente posible.

Agradecimientos

Esta investigación ha sido llevada a cabo en el entorno de trabajo de los siguientes proyectos: MODEL-CAOS (TIN2008-03582/TIN) financiado por el Ministerio de Educación y Ciencia de España, IDONEO (PAC08-0160-6141),

QUASIMODO (PAC08-0157-0668), SISTEMAS (PII2I09-0150-3135) y SEGMENT (HITO-09-138) financiados por la Viceconsejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha y FEDER y el proyecto BUSINESS (PET2008-0136) financiado por el Ministerio de Ciencia e Innovación de España.

REFERENCIAS

[1] "The Open Web Application Security Project (OWASP)",http://www.owasp.org

[2] S. T. Halkidis, N. Tsantalis, A. Chatzigeorgiou y G. Stephanides "Architectural Risk Analysis of Software Systems Based on Security Patterns" *IEEE Transactions on Dependable and Secure Computing*, pp. 129-142, 2008.

[3] C. Alexander, S. Ishikawa y M. Silverstein "*A Pattern Language: Towns, Buildings, Constructions*" Oxford University Press, 1977.

[4] E. Fernandez "Security Patterns and Secure Systems Design" en *Dependable Computing*, 2007, pp. 233-234.

[5] E. Fernandez, H. Washizaki, N. Yoshioka, A. Kubo y Y. Fukazawa "Classifying Security Patterns" en *Progress in WWW Research and Development*, 2008, pp. 342-347.

[6] E. B. Fernandez, J. Wu, M. M. Larrondo-Petrie y Y. Shao "On building secure SCADA systems using security patterns" en *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies* Oak Ridge, Tennessee: ACM, 2009.

[7] A. Maña, D. Serrano, J. F. Ruiz, A. Armenteros, B. G. N. Crespo y A. Muñoz "Development of Applications Based on Security Patterns" en *DEPEND '09. Second International Conference on Dependability*, 2009, pp. 111-116.

[8] C. Steel, R. Nagappan y R. Lai "*Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management*", Prentice Hall ed., 2005.

[9] M. Schumacher y U. Roedig "Security engineering with patterns" *PLoP 2001 Conference*, 2001.

[10] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann y P. Sommerlad "*Security Patterns: Integrating Security and Systems Engineering*", Wiley ed., 2006.

[11] K. Yskout, T. Heyman, R. Scandariato y W. Joosen "An inventory of security patterns" *Technical Report CW-469, Katholieke Universiteit Leuven, Department of Computer Science*, 2006.

[12] D. G. Rosado, C. Gutiérrez, E. Fernández-Medina y M. Piattini "Security patterns and requirements for internet-based applications" *Internet Research: Electronic Networking Applications and Policy*, 2006.

[13] B. Blakley y C. Heath "Security Design Patterns. The Open Group Security Forum" 2004.

[14] D. M. Kienzle, M. C. Elder, D. Tyree y J. Edwards-Hewitt "Security patterns repository, version 1.0" 2006.

[15] J. Garzás y M. Piattini "Object Oriented Microarchitectural Design Knowledge" *IEEE Software*, pp. 28-33, 2005.

[16] E. B. Fernandez, N. Yoshioka y H. Washizaki "Modeling Misuse Patterns" en *ARES '09. International Conference on Availability, Reliability and Security*, 2009, pp. 566-571.

[17] Z. Anwar, W. Yurcik, R. E. Johnson, M. Hafiz y R. H. Campbell "Multiple design patterns for voice over IP (VoIP) security" en *Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International*, 2006, pp. 8 pp.-492.

[18] M. Schumacher "B. Example Security Patterns and Annotations" en *Security Engineering with Patterns*, 2003, pp. 171-178.

[19] A. Sarmah, S. M. Hazarika y S. K. Sinha "Security Pattern Lattice: A Formal Model to Organize Security Patterns" en *DEXA '08. 19th International Conference on Database and Expert Systems Application*, 2008, pp. 292-296.

[20] E. Fernandez, G. Pernul y M. Larrondo-Petrie "Patterns and Pattern Diagrams for Access Control" en *Trust, Privacy and Security in Digital Business*, 2008, pp. 38-47.

[21] B. Kitchenham "Guideline for performing Systematic Literature Reviews in Software Engineering. Version 2.3" *University of Keele (Software Engineering Group, School of Computer Science and Mathematics) and Durham (Department of Conputer Science)*, 2007.

[22] E. B. Fernandez, J. C. Pelaez y M. M. Larrondo-Petrie "Security Patterns for Voice over IP Networks" en *ICCGI 2007. International Multi-Conference on Computing in the Global Information Technology*, 2007, pp. 33-33.

[23] N. A. Chavhan y S. A. Chhabria "Multiple design patterns for voice over IP security" en *Proceedings of the International Conference on Advances in Computing, Communication and Control* Mumbai, India: ACM, 2009.

[24] E. B. Fernandez y J. L. Ortega-Arjona "The Secure Pipes and Filters Pattern" en *DEXA '09. 20th International Workshop on Database and Expert Systems Application*, 2009, pp. 181-185.

[25] N. Delessy, E. B. Fernandez y M. M. Larrondo-Petrie "A Pattern Language for Identity Management" en ICCGI 2007. *International Multi-Conference on Computing in the Global Information Technology*, 2007, pp. 31-31.

[26] A. Cuevas, P. El Khoury, L. Gomez y A. Laube "Security Patterns for Capturing Encryption-Based Access Control to Sensor Data" en *SECURWARE '08. Second International Conference on Emerging Security Information, Systems and Technologies*, 2008, pp. 62-67.

[27] P. Morrison y E. B. Fernandez "The credentials pattern" en *Proceedings of the 2006 conference on Pattern languages of programs* Portland, Oregon: ACM, 2006.

[28] E. B. Fernandez y G. Pernul "Patterns for session-based access control" en *Proceedings of the 2006 conference on Pattern languages of programs* Portland, Oregon: ACM, 2006.

[29] L. L. Lobato, E. B. Fernandez y S. D. Zorzo "Patterns to Support the Development of Privacy Policies" en *ARES '09. International Conference on Availability, Reliability and Security*, 2009, pp. 744-749.

[30] S. Romanosky, A. Acquisti, J. Hong, L. F. Cranor y B. Friedman "Privacy patterns for online interactions" en *Proceedings of the 2006 conference on Pattern languages of programs* Portland, Oregon: ACM, 2006.

[31] E. Fernandez, J. Pelaez y M. Larrondo-Petrie "Attack Patterns: A New Forensic and Design Tool" en *Advances in Digital Forensics III*, 2007, pp. 345-357.

[32] T. Fischer, A. R. Sadeghi y M. Winandy "A Pattern for Secure Graphical User Interface Systems" en *DEXA '09. 20th International Workshop on Database and Expert Systems Application*, 2009, pp. 186-190.

[33] A. Sorniotti, P. El Khoury, L. Gomez, A. Cuevas y A. Laube "A Security Pattern for Untraceable Secret Handshakes" en *SECURWARE '09. Third International Conference on Emerging Security Information, Systems and Technologies*, 2009, pp. 8-14.

[34] E. B. Fernandez, M. Fonoage, M. VanHilst y M. Marta "The Secure Three-Tier Architecture Pattern" en *CISIS 2008. International Conference on Complex, Intelligent and Software Intensive Systems*, 2008, pp. 555-560.

[35] A. Aarsten, D. Brugali, G. Menga, K. Brown y R. Hirschfeld "Patterns of three-tier client server architectures" *Proceedings of the 1996 Pattern Languages of Programs (PLoP) Conference, Monticello, IL, September 1996*, 1996.

[36] O. Vogel "EuroPLoP 2001 design fest designing a three-tier architecture pattern language" *Design Fest EuroPLoP 2001, POSA3, Irsee, Germany, July 2001*, 2001.

[37] E. B. Fernandez, T. Sorgente y M. M. Larrondo-Petrie "Even more patterns for secure operating systems" en *Proceedings of the 2006 conference on Pattern languages of programs* Portland, Oregon: ACM, 2006.

[38] G. Spanoudakis, C. Kloukinas y K. Androutsopoulos "Towards security monitoring patterns" en *Proceedings of the 2007 ACM symposium on Applied computing* Seoul, Korea: ACM, 2007.

[39] E. Gamma, R. Helm, R. Johnson y J. M. Vlissides "Design Patterns: Elements of Reusable Object Oriented Software" *Addison Wesley*, 1995.

[40] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad y M. Stal. "Pattern-oriented software architecture" *Wiley*, 1996.

[41] M. Weiss y H. Mouratidis "Selecting Security Patterns that Fulfill Security Requirements" en *International Requirements Engineering, 2008. RE '08. 16th IEEE*, 2008, pp. 169-172.

Euskalert: Red Vasca de Honeypots

Urko Zurutuza, Enaitz Ezpeleta, Ignacio Arenaza, Iñaki Vélez de Mendizabal,
Jesús Lizarraga, Roberto Uribeetxeberria, Miguel Fernández
Escuela Politécnica Superior
Mondragon Unibertsitatea
Email: uzurutuza,eezpeleta,iarenaza,ivelez,jlizarraga,ruribeetxeberria,mfernandez@eps.mondragon.edu

***Abstract*—Las firmas de seguridad y especialmente los fabricantes de Antivirus dan fe del aumento exponencial de las amenazas que acechan las actividades realizadas en Internet [1], habiendo analizado más ejemplares de malware en el 2009 que en la suma de todos los años anteriores. Se constata que fabricantes de virus, gusanos, troyanos, spyware, spam etc. no realizan sus actividades maliciosas de forma aislada, sino que se trata de bandas organizadas y consolidadas [2] que buscan obtener un beneficio económico a través de sus acciones ilícitas.**

En este trabajo se presenta Euskalert, una infraestructura de máquinas trampa que recopila ataques a nivel de red y malware.

I. Introducción

A lo largo de los últimos 3 años desde la Escuela Politécnica Superior de Mondragon Unibertsitatea, en adelante MU, se ha trabajado en el proyecto Euskalert [3]. Se ha implantado una red de honeypots en la Comunidad Autónoma del País Vasco (CAPV). Los participantes alojan un sensor en su red corporativa y los datos sobre los ataques son recibidos y almacenados en nuestras instalaciones, todo ello de forma eficiente y segura. Los participantes tienen a su disposición un sitio Web donde pueden consultar libremente información y estadísticas sobre los ataques recibidos, así como compararse con otros participantes de forma anónima. Una vez la plataforma se encuentra estable y con cierta capacidad de análisis, las posibilidades en cuanto a la explotación de la información tanto a nivel de red como de aplicación se multiplican.

El malware es cualquier tipo de código diseñado específicamente con intención dañina, como por ejemplo virus, gusanos, caballos de troya o spyware. Debido al exponencial aumento del malware, y por consiguiente el beneficio económico que ciberdelincuentes obtienen mediante su uso [4], [5], el estudio, análisis y mitigación representa una necesidad prioritaria para investigadores y gobiernos.

En la actualidad, la solución principal para luchar contra el malware recae en los sistemas antivirus, basados mayoritariamente en firmas sintácticas, que caracterizan instancias conocidas de malware mediante firmas. Las firmas representan el o los bytes específicos o secuencias de instrucciones que se consideran maliciosas. Cuando al escanear un fichero se identifica este patrón, es clasificado como malware. Este método ha resultado efectivo hasta el momento, cuando las amenazas son conocidas de antemano y es la solución más extendida en el software antivirus.

El sistema Euskalert resulta muy beneficioso para obtener nuevas instancias de malware para ser analizadas. De todos modos, el método tradicional de analizar el malware implica que un analista deba realizar ciertos tests y extraer la información significativa para clasificar la muestra y desarrollar una firma específica [6]. Con el incremento percibido del número de código malicioso detectado, más de 37.000 nuevas variantes de malware al día según Panda Security, las compañías antivirus reciben miles de ficheros sospechosos que deben ser analizados y clasificados como software benigno, o al contrario malware. Por esta razón, la automatización de algunas de estas tareas de análisis y clasificación en un tiempo corto es un punto importante a tratar.

II. Estado del Arte

A. Honeypots o Máquinas Trampa

Un sistema trampa es un recurso de seguridad informática cuyo valor reside en ser explorado, atacado o puesto en compromiso [7]. Un sistema trampa no tiene ninguna función autorizada ni ningún valor productivo dentro de una red corporativa. Por tanto, un sistema trampa no debería recibir ningún tipo de tráfico. Cualquier intento de conexión con un sistema trampa es, con total seguridad, una exploración, un ataque o un intento de comprometer la máquina o el servicio que está ofreciendo [8]. Cuanto más conozcamos cómo actúan los atacantes, sus métodos y las herramientas que utilizan, mejor podremos protegernos.

Los sistemas trampa pueden clasificarse en función de varios aspectos. Por un lado, según su objetivo se pueden diferenciar entre sistemas trampa con sentido productivo (para prevención y ayuda en la respuesta en redes corporativas) y aquellos dedicados a la investigación (con el fin de recopilar información y analizarla para aprender métodos utilizados por los atacantes). Por otro lado, una de las clasificaciones más extendidas en torno a los sistemas trampa es la que hace referencia a su nivel de interacción. Los sistemas trampa de baja interacción ofrecen una baja y limitada interactividad hacia los atacantes. La mayoría de los desarrollos e implementaciones de sistemas trampa de baja interacción, no son más que simuladores de servicios y de sistemas operativos. Ejemplos de este tipo de sistemas trampa son Honeyd [9], LaBrea [10] o Honeytrap [11]. En los sistemas trampa de alta interacción la estrategia es distinta. No se simula nada, sino que se trabaja con sistemas operativos y aplicaciones reales, generalmente ejecutandose en máquinas virtuales. En este apartado se encuentran Argos [12], Minos [13], y algunos proyectos de la Honeynet Research Alliance [14]. También se podrían encontrar sistemas trampa reconocidos como de

media interacción, que también emulan servicios vulnerables, pero además dejan al propio sistema operativo gestionar las conexiones mediante sus protocolos y pila de red reales. Entre este tipo de máquinas trampa podríamos clasificar a BillyGoat [15], Nepenthes [16] o Amun Honyepot [17]. Este es el tipo de máquina trampa utilizado por los investigadores de este proyecto, por su experiencia en el desarrollo de Billy Goat (Zürich IBM Research Laboratory), y porque cumple con los requisitos necesarios ya que son capaces de gestionar la conexión llegando incluso a descargar el malware.

Por otro lado, recientemente se han propuesto otro tipo de máquinas trampa, como resultado del cambio de estrategia que se ha observado por parte de los atacantes. Los ciberdelincuentes ya no hacen uso únicamente de la capa de red para propagar el malware, sino que han sabido explotar vulnerabilidades dirigidas a aplicaciones cliente como navegadores web, pdf, jpeg, etc. Dándoles la capacidad de ejecutar código arbitrario en el lado del usuario. De este modo, en lugar de esperar de forma pasiva a que los ataques lleguen a máquinas trampa tradicionales o del lado de servidor, se han desarrollado máquinas trampa del lado de cliente, también conocidos como Honeyclients. Su función es la de rastrear un canal de comunicación, como la Web, en busca de malware. Un Honeyclient es un equipo dedicado que maneja aplicaciones específicamente instrumentadas para acceder a servidores remotos y comprobar si dichos servidores se comportan de manera maliciosa. Específicamente, estos sistemas pueden detectar proactivamente exploits contra las aplicaciones sin necesidad de firmas conocidas. Pueden hacerlo de forma automatizada, o mediante una lista de URLs que se añade manualmente o a partir de otros sistemas (links en correo spam, ...). Los ejemplos más conocidos de Honeyclients son Honeymonkey [18], MonkeySpider [19], o Capture HPC [20].

B. Infraestructuras de Máquinas Trampa Existentes para la Colección de Datos

La monitorización de tráfico no solicitado ha demostrado ser una técnica eficiente para la detección de amenazas en Internet y colección de malware. En estos últimos años, se han propuesto dos sistemas de monitorización apropiados para realizar dicha función: los sistemas trampa y los monitores de red. Ambos sistemas monitorizan direccionamiento IP no asignado de modo que cada vez que recibe una petición de conexión, ésta será considerada como sospechosa. No obstante, el nivel de interacción de estos sistemas es fundamental.

Los primeros monitores de tráfico de Internet, conocidos como Network Telescopes, Black Hole Monitors o Internet Sinks se presentan en [21]. Utilizan grandes rangos de direccionamiento IP no asignados para la recolección de información proveniente en su mayoría de gusanos informáticos tratando de propagarse, así como de fallos de configuración en equipamiento de red. Los principales Telescopios de Red son, entre otros, UCSD Network Telescope de la Universidad de California en San Diego [22] e IMS (Internet Motion Sensor) de la Universidad de Michigan [23].

Por otro lado, Existen también redes colaborativas formadas por máquinas trampa distribuidas que tratan de ocupar mayor espectro de red para obtener una vista más amplia de lo que sucede en Internet (Leurré.com/SGNET [24], NoAH [25], SURFids [26], Honeynet Project Alliance [14], o la Red Vasca de Honeypots, Euskalert [3]).

C. Técnicas de Análisis de Malware

Hasta el momento se han propuesto dos enfoques para el análisis de malware: análisis estático [27], [28] y análisis dinámico [29], [30], [31], [34], [35], [36], [37], [38], [39].

El análisis estático se realiza sin ejecutar el código malicioso, observando el código fuente o los binarios en busca de patrones sospechosos. A pesar de que algunos enfoques han obtenido buenos resultados, los autores de malware han desarrollado diferentes técnicas de ofuscación [32], como polimorfismo o cifrado, que son especialmente eficaces contra el análisis estático, tal como se presenta en el trabajo reciente [33], donde se demuestra un método para la ofuscación mediante una técnica llamada Opaque Constants.

Por otro lado, el análisis dinámico o de comportamiento [34] implica ejecutar la muestra en un entorno aislado y controlado, conocido como sandbox, para realizar un seguimiento de su comportamiento. Existen diferentes soluciones basadas en sandbox-es, utilizando técnicas para supervisar el comportamiento del malware: TTAnalyze [35] y Anubis [36] utiliza el emulador de PC Qemu para cargar un sistema operativo Windows combinado con una técnica conocida como enganche o hooking de la API. Por otra parte, CWSandbox [37] se basa también en un hooking de la API y la inyección de librerías DLL para rastrear y monitorizar todas las llamadas al sistema, y generar un informe. Norman Sanbox [38] simula una computadora completa reimplementando un sistema Windows básico. Finalmente, Wepawet [39] es un sistema de análisis dinámico de ficheros Javascript, Flash y PDF maliciosos.

En cuanto a los trabajos de clasificación automática de malware, se han propuesto diferentes técnicas. Si nos fijamos en los algoritmos de minería de datos utilizados, observamos trabajos que utilizan árboles de decisión [40], máquinas de soporte vectorial [31], y algoritmos bayesianos [41]. En cuanto al objetivo perseguido por estos trabajos, podemos diferenciar aquellos que pretenden clasificar familias de malware mediante el agrupamiento de características extraídas a partir de informes de comportamiento de malware [42]. En los mismos, se transforman los informes obtenidos del sandbox en secuencias de características, agrupando estas más tarde mediante técnicas de clustering. Esta solución tiene ciertas limitaciones, debido a la naturaleza no supervisada del clustering, con sus problemas inherentes. En otro enfoque perseguido, se trata de discriminar entre el software malicioso y el benigno, basado en la extracción de características estáticas del malware [43], o de comportamiento tanto de programas maliciosos como benignos [44].

III. EUSKALERT: RED VASCA DE HONEYPOTS

Euskalert es una red distribuida de honeypots basada en Honeynet GenIII [14]. La arquitectura desarrollada en Euskalert es la siguiente:

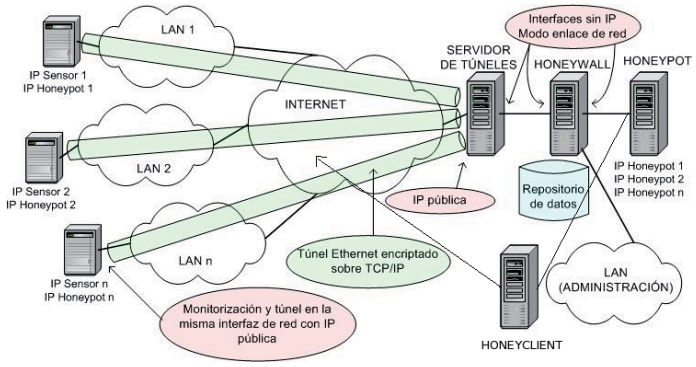


Fig. 1. Arquitectura de Euskalert, la Red Vasca de Honeypots

Como se puede observar, a la izquierda de la Figura 1 se encuentran los diferentes sensores instalados en las redes corporativas de los diferentes participantes. Cada sensor tiene permanentemente establecida una conexión encriptada (mediante diferentes redes privadas virtuales, también conocidas como VPNs) con un servidor de túneles. Éste último se encuentra en la DMZ de MU. Todo ataque o intento de propagación de ataque a los sensores son redirigidos a través de estos túneles hasta llegar al honeypot (derecha de la Figura), quien es el responsable de responder a todo intento de conexión. El tráfico también atraviesa un servidor encargado de recopilar toda la información que luego será mostrada en la plataforma Web http://www.euskalert.net.

Los elementos que componen la arquitectura son:

- **Sensores**: Pequeños dispositivos de bajo coste (Linksys NSLU2), cuyo firmware ha sido modificado para albergar un Sistema Operativo GNU/Linux con los servicios necesarios utilizando un dispositivo de almacenamiento externo USB. El software incluye Honeymole Client (The Honeynet Project), SNMP para su monitorización, el Firewall IPTables para su protección, SSH para la gestión y OpenSSL para crear un túnel cifrado con el Servidor de Túneles.
- **Servidor de Túneles**: Esta máquina es una distribución completa de GNU/Linux sobre la que se instala Honeymole Server (The Honeynet Project). Su principal cometido es el de crear túneles Ethernet encriptados (puede manejar varios simultáneamente) a partir de las peticiones hechas por los sensores Euskalert y redirigir hacia el Honeywall y honeypot las tramas encapsuladas en dichos túneles, así como remitir a los sensores las respuestas de esas conexiones. Los túneles son creados sobre la interfaz de red pública y los reenvíos de las tramas que por éstos llegan se realizan hacia el segmento de la interfaz interna.
- **Honeywall**: El Honeywall es un host que funciona como pasarela de nivel 2, uniendo el segmento de red del honeypot con el del servidor de túneles. Todo el tráfico que los sensores Euskalert envían al servidor de túneles es redirigido por este último hacia el Honeywall, posibilitándole monitorizar y registrar todos los flujos que lo atraviesan. Lleva instalado software para captura de información: Snort IDS, IPTables, Argus, p0f y Tcpdump, además de herramientas desarrolladas por MU para la extracción automática de información de origen geográfico y secuencias de puertos atacados.
- **Honeypot**: El honeypot es uno de los puntos más importantes de la infraestructura Euskalert porque representa la víctima hacia la que atacantes, gusanos y virus lanzan sus ataques. El componente software que emula máquinas y servicios se denomina honeypot, nombre que por extensión también toma el propio equipo. Euskalert hace uso del honeypot amun [17], especialmente indicado para la detección y captura de gusanos y malware en general. Como complemento suyo también se ejecuta Honeytrap, un manejador de conexiones establecidas contra puertos de servicios no emulados por Amun. La combinación de ambos proporciona la monitorización de todo el rango de puertos y la captura de la información enviada en las sesiones efectuadas contra ellos.
- **HoneyClient**: Se trata de una máquina trampa del lado de cliente con capacidad de rastrear URLs en busca de software malicioso alojado en Servidores de Internet. Puede hacer búsquedas automáticas (implica mucha carga de trabajo y poco malware), o a partir de listas predefinidas de URLs.

IV. RESULTADOS

Euskalert recopila aproximadamente una media de 4000 paquetes maliciosos al mes. La Figura 2 muestra su distribución en el mes de Marzo del 2010. En la misma, se incluye (en la línea de puntos) el número de malware que ha sido descargado en el mismo periodo, que no supera los 6 ejemplares por día:

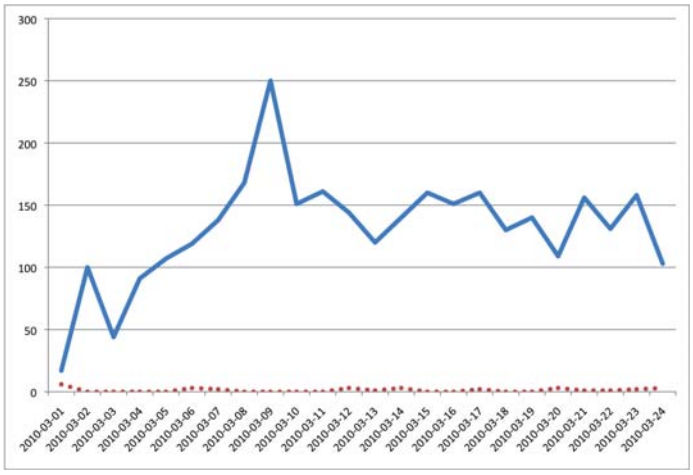


Fig. 2. Distribución diaria de ataques a la plataforma Euskalert

En cuanto al contenido del tráfico, la tabla I muestra los servicios más atacados en la misma muestra anterior:

Puerto Destino	N. Paquetes	Descripción
445	2091	Microsoft-DS compartición de ficheros
8	412	ICMP echo
5061	217	Inicio de sesión sobre TLS
135	202	DCE endpoint resolution de Microsoft
22	197	ssh, Secure Shell
139	152	NetBIOS Servicio de sesiones
80	139	HTTP
443	135	Protocolo HTTP sobre TLS/SSL
1433	76	Microsoft-SQL-Server
1521	76	Servidor de Base de Datos Oracle

TABLE I
SERVICIOS (PUERTOS) MÁS ATACADOS

Algunos resultados, como la geolocalización de los atacantes, los ataques detectados por el IDS Snort, y los servicios y protocolos atacados se publican de forma automática en la dirección http://www.euskalert.net, tal y como se muestra en la Figura 3.

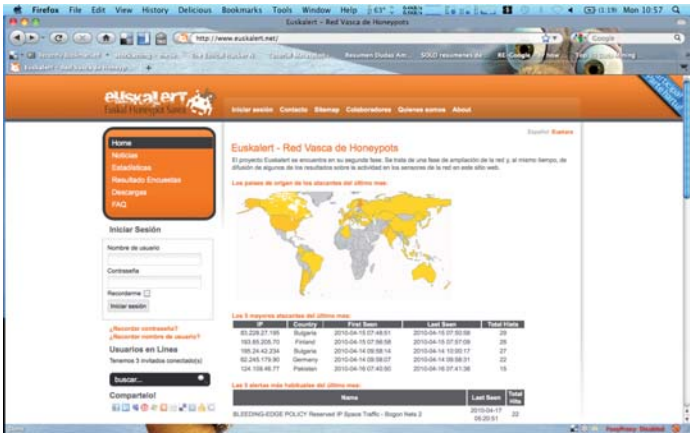


Fig. 3. Interfaz web de la plataforma Euskalert

En cuanto a ataques específicos, a modo de ejemplo se muestran por un lado intentos de acceso a nivel FTP, y por el otro intentos de ataques automáticos vía HTTP a diferentes aplicaciones Web conocidas:

```
RMD sarcaxxo
PASS .
USER administrator
PASS NULL
MKD 090713182104p
CWD /public/
PASS Pgpuser@home.com
USER anonymous
PASS ftp@example.com
PWD EPSV
```

Fig. 4. Intentos de acceso vía FTP

```
GET /twiki/bin/statistics HTTP/1.1 Accept: */*
Accept-Language: en-us Accept-Encoding: gzip,
deflate User-Agent: Toata dragostea mea pentru
diavola Host: ***.***.***.49 Connection: Close

OPTIONS / HTTP/1.1 translate: f User-Agent:
Microsoft-WebDAV-MiniRedir/5.1.2600 Host:
***.***.***.61 Content-Length: 0 Connection:
Keep-Alive

GET /roundcube/program/js/list.js HTTP/1.1 Accept:
*/* Accept-Language:en-us Accept-Encoding: gzip,
deflate User-Agent: Toata dragostea mea pentru
diavola Host: ***.***.***.49 Connection: Close

GET/unauthenticated/..%01/..%01/..%01/..%01/..%01/
..%01/..%01/..%01/etc/passwd HTTP/1.1
Accept: */* Accept-Language: en-us Accept-Encoding:
gzip, deflateUser-Agent: Toata dragostea mea pentru
diavola Host: ***.***.***.49Connection: Close

POST /webmail/bin/html2text.php HTTP/1.0 Host:
***.***.***.69 User-Agent:Mozilla/4.0 (compatible;
MSIE 7.0; Windows NT 5.1; InfoPath.2)
Content-length: 54
Accept:ZWNobygiU3VjY2VlZGVkISA6KSlcclxuXHJcbi4iKTsK
<b>{${EVAL(BASE64_DECODE($_SERVER[HTTP_ACCEPT]))}}
</b>
```

Fig. 5. Ataques / pruebas vía HTTP

V. CONCLUSIONES Y PRÓXIMOS PASOS

El objetivo prioritario de Euskalert es el de mantener una red de máquinas trampa (de distinta índole) que sirva como observatorio de activad maliciosa en la red. Esto permite disponer de un sistema de alerta temprana para empresas e instituciones y de un almacén centralizado de ataques para utilizar en proyectos de investigación científica en el área de la seguridad telemática. El sistema actual se encuentra estable y recopila ataques de distintos tipos, así como malware de forma automática. El siguiente paso lógico será el de analizar los ataques que llegan y así aprender de nuevas tendencias, técnicas etc. de los atacantes, para poder investigar métodos para su defensa.

Para completar el sistema actual, se plantean los siguientes pasos:

- Obtener mayor volumen de datos aumentando el número de sensores de la red distribuida de máquinas trampa.
- Desarrollar un método para la investigación de amenazas en entornos Web 2.0, automatizando servicios como la Mensajería Instantánea (Instant Messaging) y como la herramienta Twitter.
- Desarrollar un método para emular aplicaciones Web y así recoger ataques (Cross Site Scripting, SQL Injection,..) dirigidos a aplicaciones específicas de http.
- Establecer y desarrollar una metodología de minería de datos para identificar de forma automática el malware recibido por el sistema, utilizando información de análisis estático y dinámico del malware, proporcionado por las empresas colaboradoras.

- Diseñar y desarrollar un sistema Business Intelligence e integrarlo en el sitio Web de Euskalert.

AGRADECIMIENTOS

El proyecto Euskalert está cofinanciado por la Excelentísima Diputación Foral de Guipúzcoa mediante el Programa "Red Guipuzcoana de Ciencia, Tecnología e Innovación". Este trabajo ha sido desarrollado por el grupo de Sistemas Empotrados con el apoyo del Departamento de Educación, Universidades e Investigación del Gobierno Vasco.

REFERENCES

[1] Panda Security, "Informe Anual de Malware PandaLabs", disponible en http://www.pandasecurity.com/img/enc/Informe_Anual_Pandalabs_2009.pdf, Enero 2010.

[2] B. Schneier. Organized cybercrime. Disponible en http://www.schneier.com/blog/archives/2006/09/organized_cyber.html, Septiembre 2006.

[3] Euskalert, Red Vasca de Honeypots. Disponible en http://www.euskalert.net, Enero 2010.

[4] Malware Report: The Economic Impact of Viruses, Spyware, Adware, Botnets, and Other Malicious Code. Disponible en http://www.computereconomics.com/page.cfm?name=Malware%20Report Enero 2010.

[5] El poder del dinero, Análisis de Amenazas Globales, vol. 1, number 1, page 13. www.mcafee.com/us/local_content/white_papers/\\threat_center/mcafee_sage_v11_en.pdf, Enero 2010.

[6] Exploring multiple execution paths for malware analysis. Moser, A. and Kruegel, C. and Kirda, E. En actas de 28 IEEE Symposium on Security and Privacy. (páginas 231–245). 2007.

[7] Charles, K. Decoy Sytems. International Journal of Digital Evidence, 2004, Volume 2, Issue 3, Enero de 2004.

[8] Gallego, E., López de Vergara, J. Honeynets: Aprendiendo del Atacante. IX Congreso Nacional de Internet, Telecomunicaciones y Movilidad, Febrero de 2004.

[9] N. Provos. A virtual honeypot framework. En actas del 12 USENIX Security Symposium, páginas 1–14, Agosto 2004.

[10] LaBrea. Disponible en http://labrea.sf.net/labrea-info.html, Enero 2010.

[11] T. Werner. Honeytrap. Disponible en http://honeytrap.carnivore.it, Enero 2010.

[12] The Argos Development Team. Disponible en http://www.few.vu.nl/argos, Enero 2010.

[13] J. Crandall, F. Chong, and S. Wu. Minos: Architectural Support for Protecting Control Data. In Transactions on Architecture and Code Optimization (TACO). Volume 3, Issue 4, Diciembre 2006.

[14] The Honeynet Project. Disponible en http://www.honeynet.org/misc/project.html, Enero 2010.

[15] J. Riordan, D. Zamboni, and Y. Duponchel. Building and deploying billy goat, a worm detection system. En actas de 18 Annual FIRST Conference, 2006.

[16] The nepenthes platform: An efficient approach to collect malware. Baecher, P. and Koetter, M. and Holz, T. and Dornseif, M. and Freiling, F. Lecture Notes in Computer Science (Springer). Volume 4219, página 165, 2006.

[17] Amun: Python Honeypot. Disponible en http://amunhoney.sourceforge.net/index.php, Enero 2010.

[18] Y.Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, and S. King. Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities. En actas de Network and Distributed System Security (NDSS) Symposium, páginas 3958, Febrero 2006.

[19] Piotr Kijewski, Carol Overes and Rogier Spoor. The HoneySpider Network fighting client-side threats. En actas de 20 Annual FIRST Conference, 2008.

[20] The Honeynet Project. Página de inicio de Capture-HPC. Disponible en https://projects.honeynet.org/capture-hpc, Enero 2010.

[21] D. Moore, G. M. Voelker, and S. Savage, Inferring internet Denial-of-Service activity, En actas de Usenix Security Symposium, 2001, pp. 9–22.

[22] U. N. Telescope, UCSD network telescope, 2005. Disponible en http://www.caida.org/data/passive/network_telescope.xml.

[23] U. of Michigan, Internet Motion Sensor. Disponible en http://ims.eecs.umich.edu/index.html, Octubre de 2005.

[24] C. Leita and M. Dacier, SGNET: a worldwide deployable framework to support the analysis of malware threat models, en EDCC 2008, 7th European Dependable Computing Conference, May 7–9, 2008, Kaunas, Lithuania, Mayo 2008.

[25] E. Markatos and K. Anagnostakis, NoAH: A European Network of Affined Honeypots for Cyber-Attack Tracking and Alerting, The Parliament Magazine, Issue 262, 3 Marzo 2008.

[26] SURFids, open source Distributed Intrusion Detection System based on passive sensors. Disponible en http://ids.surfnet.nl/.

[27] Static Analysis of Executables to Detect Malicious Patterns. Mihai Christodorescu and Somesh Jha. En actas de 12 USENIX Security Symposium. (páginas 169–186). 2003.

[28] Digital genome mapping–advanced binary malware analysis. Carrera, E. and Erdelyi, G. En actas de 14 Virus Bulletin Conference. (páginas 187–197). 2004.

[29] Christodorescu, M. and Jha, S. and Kruegel, C. Mining specifications of malicious behavior. En actas de 6 joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering. (páginas 5–14). 2007.

[30] Lee, T. and Mody, J.J. Behavioral classification. En actas de 15 Annual Conference of the European Institute for Computer Antivirus Research (EICAR). 2006.

[31] Rieck, K. and Holz, T. and Willems, C. and Dussel, P. and Laskov, P. Learning and Classification of Malware Behavior. Lecture Notes in Computer Science (Springer). (páginas 108–125). 2008.

[32] Szor, P. The art of computer virus research and defense. Addison-Wesley Professional. 2005.

[33] Moser, A. and Kruegel, C. and Kirda, E. Limits of static analysis for malware detection. En actas de 23 Annual Computer Security Applications Conference (ACSAC). (páginas 421–430). 2007.

[34] Ulrich Bayer, Andreas Moser, Christopher Kruegel, and Engin Kirda, Dynamic Analysis of Malicious Code, Journal in Computer Virology, Springer Computer Science. 2006.

[35] Ulrich Bayer, Christopher Kruegel, and Engin Kirda. TTAnalyze: A tool for analyzing malware. En actas de 15 Annual Conference of the European Institute for Computer Antivirus Research (EICAR). 2006.

[36] Ulrich Bayer, Imam Habibi, Davide Balzarotti, Engin Kirda, and Christopher Kruegel, Insights Into Current Malware Behavior, 2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), Boston, MA, Abril de 2009.

[37] Willems, C. and Holz, T. and Freiling, F. Toward automated dynamic malware analysis using cwsandbox. IEEE Security & Privacy. Volume 5, number 2, páginas 32–39. 2007.

[38] Norman sandbox whitepaper. Norman Solutions. Disponible en http://download.norman.no/whitepapers/whitepaper_Norman_SandBox.pdf. Enero 2010.

[39] Wepawet. Disponible en http://wepawet.cs.ucsb.edu/index.php. Enero del 2010.

[40] Moskovitch, R., Stopel, D., Feher, C., Nissim, N. and Elovici, Y. Unknown malcode detection via text categorization and the imbalance problem. En actas de 6 IEEE International Conference on Intelligence and Security Informatics (ISI). (páginas 156–161). 2008.

[41] Kolter, J.Z. and Maloof, M.A. Learning to detect malicious executables in the wild. En actas de 10 ACM SIGKDD international conference on Knowledge discovery and data mining. (páginas 470–478). 2004.

[42] Ulrich Bayer, Paolo Milani, Clemens Hlauschek, Christopher Kruegel, and Engin Kirda, Scalable, Behavior-Based Malware Clustering, 16th Annual Network and Distributed System Security Symposium (NDSS 2009), San Diego, Febrero de 2009.

[43] M. G. Schultz, E. Eskin, and E. Zadok. Data mining methods for detection of new malicious executables. En actas de IEEE Symposium on Security and Privacy, páginas 38–49, 2001.

[44] Christodorescu, M. and Jha, S. and Kruegel, C. Mining specifications of malicious behavior. En actas de 6 joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering. (páginas 5–14). 2007.

A real-time stress detection system based on GMM for Intrusion Detection

A. de Santos Sierra, C. Sánchez Ávila
G. Bailador del Pozo, J. Guerra Casanova, V. Jara Vera
Grupo de Biometría y Tratamiento Numérico de la información
Centro de Domótica Integral
Email: {alberto,csa,gbailador,jguerra,vjara}@cedint.upm.es

***Abstract*—Main goal of biometric systems relies on avoiding intrusion acceptance. However, most extended biometric techniques lack of skills when providing information on individual intentions. Most intrusions imply an arousal in the human physiological response, better known as stress response. Therefore, stress detection is strongly related to intrusion detection. This paper attempts to provide a stress detection scheme based on Gaussian Mixture Models, elucidating to what extend an individual is under stress. Furthermore, results come up to stress detection rates of $96.2 \pm 0.2\%$ and non-stress detection rates of $86.5 \pm 0.5\%$, which means that the system can detect properly the degree of stress on an individual with competitive results if compared to the literature.**

I. Introduction

Intrusion detection attempts to identify malicious behavior of a certain user within a network or system [1]. More in detail, Intrusion Detection and Prevention Systems (IDPS) focus on detecting and identifying incidents, logging any related information and avoiding conflictive and insecure situations. Several approaches have been already proposed in the literature providing different strategies to tackle with this problem. For instance, Fast Inductive Learning [2], Support Vector Machine [3] or Neural Networks [4] among others [5], [6] provide good results in intrusion detection.

Besides, biometrics have become recently into an important topic of interest in relation to intrusion detection, since biometric techniques allow to identify/authenticate a user in a precise manner, and therefore, ensuring the fact that associating actions to individuals may increase the detection of intrusive activities [7], [8]. These biometric techniques are based concretely on behavioral biometrics [9], [10], assuming that a certain user usually behaves in the same manner in a system. Any uncommon behavior would indicate a possible intrusive and malicious activity.

Regarding biometrics, this paper proposes to elucidate intrusive behaviors by means of stress detection. This approach implies to detect the arousal emotions of an individual, or more specifically, the human stress response. Stress detection is known to provide real-time information on the emotional state-of-mind of an individual with almost non-invasive acquisition devices [11], [12], being a very suitable technique for intrusion detection. Stress detection exceeds previous biometric approaches based on two reasons: firstly, stress detection is more related to human intentions than behavioral biometrics [13], [14]; secondly, stress detection is a reliable and trustworthy technique, since the human stress response cannot be forced or disguised, in the same way an individual cannot stop his/her heart by his/her own [15], [16], [17].

The relation between stress and malicious actions is obvious: high levels of stress indicates uncommon or abnormals situations in where an individual could pretend commit a malicious actions, or the environment is affecting the individual (a robbery, for instance).

However, how is it possible to measure human stress? Most common approaches in stress detection consider physiological signals [18] as the most suitable bio-marker in terms of stress arousal. This paper focuses on two specific physiological signals, namely Galvanic Skin Response (GSR) [19], [20] and Heart Rate (HR) [21], [15]. Furthermore, this paper proposes the use of Gaussian Mixture Models (GMM) [22], as a proper method to deal with the information provided by previous physiological signals.

Moreover, its simplicity and non-invasiveness make of this approach a suitable scheme to be easily embedded in general current accessing systems.

Results direct attention towards the fact that stress detection can be implemented for real-time applications with a high performance.

The layout of this paper is as follows: Section II presents how the database was acquired to validate the algorithms, stating the mathematical model of the algorithm in Section III. Results will be provided under Section IV, yielding to the Conclusions and Future Work presented in Section V.

II. Data Acquisition

Acquiring precise data is one of the main problems of stress detection. A database to validate the implemented algorithms must include two states: a relax and a stress state, in order to compare to what extend the system is able to distinguish between a stressing and a normal situation, or in other words, between and intrusive or a non-intrusive scene. Since the stress arousal cannot be controlled by an individual [15], [16], then the stress stimuli must be induced and provoked. Thus, a experimental setup was carried out in order to arise stress on individuals, based on two stressing tasks: Hyperventilation (HV) and Talk Preparation (TP), [23], [24], which have been extensively used to provoke stress on individuals with very

satisfactory results. Firstly, HV represents a kind of breath, exceeding standard metabolic demands. Secondly, TP is a very anxiogenic task which ensures a positive valence in terms of stress response. Furthermore, despite of provoking similar responses on humans, HV does not produce a significant increment in anxiety intensity, and therefore, a more anxiogenic task, TP, is required.

These experiments were carried out in a Faraday room in the Human Psychology Laboratory from Psychology Faculty of Complutense University of Madrid (UCM), provided with electromagnetic, thermal and acoustic insulation. As previously stated, HR and GSR are collected during the experiments from each participant, having sensors attached to wrists, fingers and ankles of the individual [25]. The device proposed is I-330-C2 PHYSIOLAB (J &J Engineering) able to process 6 channels. Finally, the number of individuals was 80 female individuals, with ages from 19 to 32 years, and an average of 21.8 years and a deviation of 2.15 years.

A. Experimental Setup

The experimental setup consisted of a set of task (stressing and non-stressing) in order to collect the data from HR and GSR signals in each situation. The procedure is described as follows in four steps:

- First, a relaxing step was carried out at the beginning asking kindly the participant to relax some minutes seated in a comfortable chair. After some minutes, HR and GSR signals were recorded for 120 seconds. This step is called Base Line 1, BL1, and corresponds to a non-stressing task.
- Afterwards, the participant is required to deeply breath in short periods of time, indicated by the experimenter. This procedure was carried out till the participant clearly felt changes in his or her corporal sensation. At this moment, HR and GSR are collected again for a period of time of 90 seconds. This step is called Hyperventilation, HV, and corresponds to a stressing task.
- Third task consisted of asking the participant to prepare a short speech on a certain topic during a short period of time, being recorded afterwards. Before recording, HR and GSR were collected during 90 seconds. Finally, the participant was said not to be recorded. This step also corresponds to a stressing task, and is called TP.
- Finally, the participant is said to relax, since the experiment have come to an end, recording in this period the physiological signals during 120 seconds. This step is defined as Base Line 2, BL2, and there is no certainty on the extend of stress of this task (stressing or non-stressing), but it can be considered as a post-stress task [23], [24].

For the sake of task order independence, HV and TP tasks were swapped for half of the database population.

Besides, several psychological tests were carried out together with the signal collection. The explanation of this test is beyond the scope of this paper, but represents a proper manner for detecting a subjective degree of stress provided by the individual [26], [27].

B. Database discussion

This data acquisition collects one unique sample of an individual of both physiological signals HR and GSR. Notice that this psychological experiment is far from being repeatable, due to the fact that the specific tasks previously described (BL1, HV, TP and BL2) require a component of surprise and unexpectedness. In other words, if an individual carries out again the same tasks, then the participant could be prepared to come through the task, and what is more, the response of her or his physiological signals will not be certainly the same. This fact justifies that only one sample is considered for each individual.

Furthermore, the stress response of an individual is slightly different depending on the stressing task, but similar enough to assume that a system trained with this database could detect stress on a real-time application [23], [24]. On the other hand, despite of having different responses among males and females, the behavior of the human stress response is similar in both genus, and therefore, the implemented system might be able to detect stress also in male individuals [28], [29], [30].

However, the algorithm responsible for detecting stress has been implemented independently of these likely drawbacks, since it considers how an individual behaves under both stress and relax situation, providing in theory more independence from database population.

III. Stress Detection

Human stress response attends the physiological demands under abnormal situations. This response is different for each individual, and therefore defining a unique criteria for stress detection lacks of interest. This scheme proposes a stress template for each individual, gathering the behavior of physiological signals (GSR and HR) under both stressing and non-stressing situations. This template is created based on data acquired from an individual, extracted by a mixture of Gaussian functions (GMM) [22].

Subsequent sections will provide an overview on Gaussian Mixture Model (Section III-A), how the template is extracted and which schemes (Section III-B) are proposed to be assessed under the Results section (Section IV).

A. Gaussian Mixture Model

Four different acquisitions from an individual were collected during the experiments.

Let $\mathbf{x}$ be a two-dimensional observation describing a sample of both GSR and HR. The probability density function of $\mathbf{x}$ in the finite mixture form is expressed in Eq. 1 and Eq. 2,

$$p(\mathbf{x};\phi_c) = \sum_{i=1}^{K} \pi_i g(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i) \tag{1}$$

$$g(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i) = \frac{1}{2\pi|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)} \tag{2}$$

where K is the number of mixtures, the parameter $\phi_c = \{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^{K}$ consists of the mixture weight π_i ($\Sigma\pi_i = 1$), the mean vector $\boldsymbol{\mu}_i$ and the covariance matrix $\boldsymbol{\Sigma}_i$ of the ith Gaussian component $\forall i = 1, 2, \ldots, K$, in the c class. In fact, Eq. 2 is a specific case with $d = 2$ [22], [31], since there are only two physiological signals, HR and GSR.

The parameters represented by ϕ_c are estimated by applying the Expectation Maximization (EM) algorithm [31]. Let $\{\mathbf{x}^t\}_{t=1}^{N}$ be the training samples, then EM algorithm finds

$$\phi_c^* = \text{argmax } \Pi_{t=1}^{N} P(\mathbf{x}^t|\phi_c) \tag{3}$$

Section III-B will define which values of K (number of mixture per class) and c (number of classes) are more suitable aiming maximum success in stress detection.

B. Template extraction

Considering previous mathematical model, there exist several possibilities for a template extraction. There exist four possible classes BL1, HV, TP and BL2, however, BL2 is difficult to define as a stressing/non-stressing state, since it corresponds to a post-stress state, and therefore, there is no possible certainty in which degree of stress involves such a state [23], [24]. Thereby, BL2 will not be considered in this experiments, and its analysis remain as future work. This approach will consider three possible templates, whose performance will be compared under the Results Section IV.

- Three possible classes, BL1, HV and TP. Data from stages BL1, HV and TP is gathered and divided into three classes. Each class will be modeled with a unique Gaussian, i.e. K=1 and c=3. Figure 1 represents the output of the GMM model. Notice how three groups are distinguished, namely BL1, HV and TP.
- Two possible classes, Stress (HV and TP) and non Stress (BL1). Stages HV and TP are gathered, since they correspond to the same state-of-mind [23], i.e. K=1 and c=2. Figure 2 represents the fact of requiring only two possibilities: stress and non-stress.
- Modeling each state as one class. This approach differs from first approach (three possible classes) in the fact that each acquisition from each stage is modeled independently, i.e. K = 1, and c = 1.

IV. Results

Since there is only one sample for each user, the data acquired must be split into several parts in order to obtain enough samples for training and testing the proposed approaches. However, is it possible to divide the signal into different parts and consider those parts as independent samples? The question is difficult to be answered, but there exists previous work on this topic [12], concluding that it is possible and reasonable to state such an assumption.

Thereby, part of the data acquired of a participant is divided into two slots: First part is used to create the template, and the other part is used to test the proposed approaches. Considering that the device used in this paper (I-330-C2 PHYSIOLAB) provides an output each second of both HR and GSR, then each point of the data represent one second, and therefore, a whole data acquisition consists of 420 points.

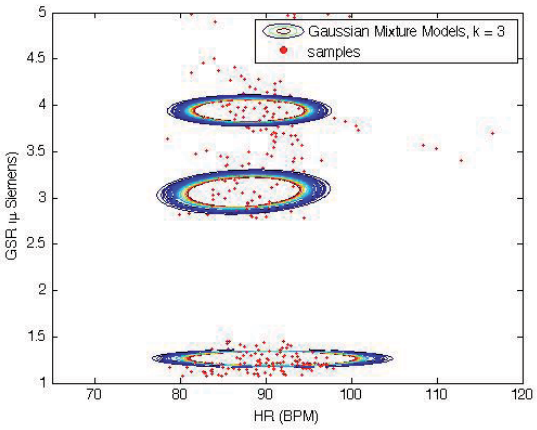


Fig. 1. GMM with c = 3. Notice how the algorithm is able to detect properly three different regions (BL1, HR, TP).

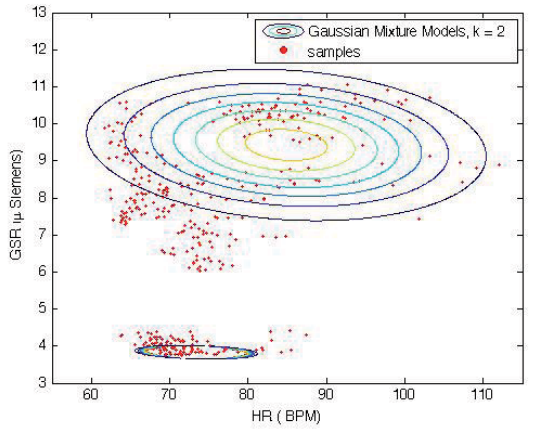


Fig. 2. GMM with c = 2. Notice how the algorithm is able to detect properly two different regions (Stress and relax).

The fact of dividing the whole signal into two parts yields to the inclusion of two more parameters, namely α_t (time to create the template) and α_{acq} (time to acquired data and decide to what extend an individual is under stress). For the sake of simplicity, α_t and α_{acq} values are stated to 3,5,7,10 and 15 seconds only.

Finally, the performance of the stress detection system will be evaluated by two parameters: True Stress Detection (TSD) which will provide a degree in detecting stress, and True Non-Stress Detection (TNSD), similar to TSD, but focused on detecting relax states. Notice that a stress detection system must reach a compromise between these two values, achieving the highest values for both parameters. In other words, a rate of 100% in detecting correctly TSD could be reached but at the expense of TNSD which could be extensively decreased. Therefore, a compromise between TSD and TNSD must be achieved.

	$c=3$	$c=2$	$c=1$
TSD	95.1± 0.2 % $\alpha_{acq}=5s$ $\alpha_t=10s$	93.4 ± 0.2 % $\alpha_{acq}=7s$ $\alpha_t=7s$	96.2 ±0.2% $\alpha_{acq}=5s$ $\alpha_t=15s$
TNSD	86.3± 0.4 % $\alpha_{acq}=5s$ $\alpha_t=7s$	82.3 ± 0.7 % $\alpha_{acq}=5s$ $\alpha_t=7s$	86.5 ±0.5% $\alpha_{acq}=3s$ $\alpha_t=10s$

TABLE I
STRESS DETECTION ACCURACY (TSD AND TNSD) OBTAINED IN RELATION TO TEMPORAL PARAMETERS α_{acq} AND α_t.

A. Temporal parameters

The temporal parameters measure the time required to detect stress. In fact, a high value in α_{aqc} might indicate that this system is not suitable for real-time applications. Therefore, the relation between α_{acq}, α_t and the performance rates (TSD and TNSD) of each approach must be studied, in order to obtain the combination of previous temporal parameters which minimize the stress detection rates. Obviously, there is also a compromise in here: the more time the system takes to detect stress, the higher the performance, but to what extend is the system able to detect stress by minimizing α_{acq} and α_t?

B. Stress Detection accuracy

Finally, the overall performance for each approach with optimal values of α_{acq} and α_t is provided in Table I.

The system is able to detect stress with 96.2 ±0.2% of accuracy, and relax state with 86.5 ±0.5% both for the third approach based on one class for each stage. This results are promising if compared to other results available in the literature considering physiological signals, which obtained rates of 97.4% using an acquisition time of 5 minutes [32], 79.5-96.6% [33], 85-96% [25], 75-85% [7], 76% [17] and 60-78% [34].

V. CONCLUSIONS

A stress detection system based on GMM has been presented with promising results. Rates of True Stress Detection of $96.2 \pm 0.2\%$ yields to the conclusion of a precise stress detection. Furthermore, the required time to elicit this results are suitable for real-time applications: a time of 5 seconds to extract the information on the state-of-mind of the user.

These results yield the conclusion that this system is suitable for scenarios of intrusion detection. Moreover, this stress detection system can present an improvement in terms of security for current biometric applications, since the system is very easy to be embedded on biometric systems.

Finally, an improvement of this technique and research with different methods remain as future work, together with an embedded implementation of this system, integrating both software and required hardware.

REFERENCES

[1] NIST, "Guide to intrusion detection and prevention systems, nist csrc special publication sp 800-94," 2007.

[2] W. Yang, W. Wan, L. Guo, and L.-J. Zhang, "An efficient intrusion detection model based on fast inductive learning," in *Machine Learning and Cybernetics, 2007 International Conference on*, vol. 6, Aug. 2007.

[3] X. Bao, T. Xu, and H. Hou, "Network intrusion detection based on support vector machine," in *Management and Service Science, 2009. MASS '09. International Conference on*, Sept. 2009, pp. 1–4.

[4] G. Liu and X. Wang, "An integrated intrusion detection system by using multiple neural networks," in *Cybernetics and Intelligent Systems, 2008 IEEE Conference on*, Sept. 2008, pp. 22–27.

[5] J. Liu and L. Li, "A distributed intrusion detection system based on agents," in *Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on*, vol. 1, Dec. 2008, pp. 553–557.

[6] J. Wang, Q. Yang, and D. Ren, "An intrusion detection algorithm based on decision tree technology," in *Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on*, vol. 2, July 2009, pp. 333–335.

[7] J. Liu, F. Yu, C.-H. Lung, and H. Tang, "Optimal combined intrusion detection and biometric-based continuous authentication in high security mobile ad hoc networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 2, pp. 806–815, Feb. 2009.

[8] A. A. E. Ahmed and I. Traoré, "Detecting computer intrusions using behavioral biometrics," in *PST*, 2005.

[9] S. Giroux, R. Wachowiak-Smolikova, and M. Wachowiak, "Keypress interval timing ratios as behavioral biometrics for authentication in computer security," in *Networked Digital Technologies, 2009. NDT '09. First International Conference on*, July 2009, pp. 195–200.

[10] R. Moskovitch, C. Feher, A. Messerman, N. Kirschnick, T. Mustafic, A. Camtepe, B. Lohlein, U. Heister, S. Moller, L. Rokach, and Y. Elovici, "Identity theft, computers and behavioral biometrics," in *Intelligence and Security Informatics, 2009. ISI '09. IEEE International Conference on*, June 2009, pp. 155–160.

[11] J. Zhai, A. Barreto, C. Chin, and C. Li, "Realization of stress detection using psychophysiological signals for improvement of human-computer interactions," in *SoutheastCon, 2005. Proceedings.*, April 2005, pp. 415–420.

[12] W. Picard, J. A. Healey, and J. A. Healey, "Wearable and automotive systems for affect recognition from physiology," MIT, Tech. Rep., 2000.

[13] J. Zhai and A. Barreto, "Stress detection in computer users through non-invasive monitoring of physiological signals," *Biomedical Science Instrumentation*, vol. 42, pp. 495–500, 2006.

[14] S. Yanushkevich, P. Wang, M. Gavrilova, M. Nixon, and S. Srihari, *Image pattern recognition: synthesis and analysis in biometrics*. World Scientific Pub Co Inc, 2007.

[15] J. Choi and R. Gutierrez-Osuna, "Using heart rate monitors to detect mental stress," in *BSN '09: Proceedings of the 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 219–223.

[16] R. M. Sapolsky, "Individual differences and the stress response: studies of a wild primate," *Adv. Exp. Med. Biol.*, pp. 399–411, 1988.

[17] D. Kulic and E. Croft, "Anxiety detection during human-robot interaction," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005).*, Aug. 2005, pp. 616–621.

[18] O. Villon and C. Lisetti, "Toward recognizing individual's subjective emotion from physiological signals in practical application," in *Computer-Based Medical Systems, 2007. CBMS '07. Twentieth IEEE International Symposium on*, Maribor,, Jun. 2007, pp. 357–362.

[19] M. M. Moore and U. Dua, "A galvanic skin response interface for people with severe motor disabilities," *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2004*, pp. 48–54, 2004.

[20] N. D. Ahuja, A. K. Agarwal, N. M. Mahajan, N. H. Mehta, and H. N. Kapadia, "Gsr and hrv: its application in clinical diagnosis," in *Computer-Based Medical Systems, 2003. Proceedings. 16th IEEE Symposium*, 2003, pp. 279–283.

[21] E. Jovanov, A. O'Donnell Lords, D. Raskovic, P. G. Cox, R. Adhami, and F. Andrasik, "Stress monitoring using a distributed wireless intelligent sensor system," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 22, no. 3, pp. 49–55, 2003.

[22] G. J. McLachlan and K. E. Basford, *Mixture models. Inference and applications to clustering*. Statistics: Textbooks and Monographs, New York: Dekker, 1988.

[23] A. Cano-Vindel, J. J. Miguel-Tobal, H. Gonzalez-Ordi, and I. Iruarrizaga-Diez, "Hyperventilation and anxiety experience," *Anxiety and stress*, vol. 13, no. 2–3, pp. 291–302, 2007.

[24] M. J. Zvolensky and G. H. Eifert, "A review of psychological factors/processess affecting anxious responding during voluntary hyperventilation and inhalations of carbon dioxide-enriched," *Clinical Psychological Review*, vol. 21, pp. 375–400, 2001.
[25] H. Cai and Y. Lin, in *An experiment to non-intrusively collect physiological parameters towards driver state detection*. Academic Press, 2007.
[26] J. J. Miguel-Tobal and A. Cano-Vindel, "Isra: Inventory of situations and responses of anxiety," 2002.
[27] R. A. Peterson and S. Reiss, "Anxiety sensitivity index manual," *International Diagnostic Systems*, 1992.
[28] O. M. Augustin, F. S. de Medina, and F. J. S. de Medina, "Effect of psychogenic stress on gastrointestinal function," *Journal of physiology and biochemistry*, vol. 56, no. 3, pp. 259–265, 2000.
[29] D. A. Padgett and R. Glaser, "How stress influences the immune response," *Trends in Immunology*, vol. 24, no. 8, pp. 444 – 448, 2003.
[30] M. LeMay, F. Layton, and D. J. Townsend, "A model of human response to workload stress," *Bullet of the Psychonomic Society*, vol. 28, no. 6, pp. 547–550, 1990.
[31] J. H. Wolfe, "Pattern clustering by multivariate mixture analysis," *Multivariate Behavioral Research*, vol. 5, pp. 329–350, 1970.
[32] J. A. Healey and R. W. Picard, "Detecting stress during real-world driving tasks using physiological sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, pp. 156–166, Jun. 2005.
[33] J. Wagner, N. J. Kim, and E. Andre, "From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification," *IEEE Computer Society*, pp. 940–943, 2005.
[34] M. S. Sharawi, M. Shibli, and M. I. Sharawi, "Design and implementation of a human stress detection system: A biomechanics approach," in *5th International Symposium on Mechatronics and Its Applications, 2008. ISMA 2008.*, Amman, May 2008, pp. 1–5.

Security analysis of JXME-Proxyless version

Marc Domingo-Prieto, Joan Arnedo-Moreno
Estudis d'Informàtica, Multimèdia i Telecomunicació
Universitat Oberta de Catalunya
Barcelona, Spain
{mdomingopr, jarnedo}@uoc.edu

Jordi Herrera-Joancomartí
Escola d'Enginyeria
Universitat Autònoma de Barcelona
Campus de Bellaterra, Spain
jherrera@deic.uab.cat

***Abstract*—JXME is the JXTA specification for mobile devices using J2ME. Two different flavors of JXME implementation are available, each one specific for a particular set of devices, according to their capabilities. The main value of JXME is its simplicity to create peer-to-peer (P2P) applications in limited devices. In addition to assessing JXME functionalities, it is also important to realize the default security level provided. This paper presents a brief analysis of the current state of security in JXME, focusing on the JXME-Proxyless version, identifies existing vulnerabilities and proposes further improvements in this field.**

I. Introduction

Peer-to-peer (P2P) networks allow peers to provide and consume services in a collaborative way. Examples of these services are content sharing, processing and messaging. In this kind of network, it is assumed that all peers have equivalent capabilities, as well as a high degree of decentralization and autonomy.

P2P technology, that has been widely used in traditional wired network environments, is now moving to the mobile paradigm [1] since new wireless technologies are becoming more available (WLAN, 3G, 3.5G,...) and more powerful handheld devices (like smart phones or mobile Internet devices, MID) have been developed. However, the massive deployment of P2P mobile applications may depend on the tools available for developing such applications in a transparent way.

There are different platforms that allow programmers to develop mobile P2P applications ([2], [3], [4]). One of these platforms is JXME [4], a set of open protocols specifications that enables the creation and deployment of P2P networks over mobile devices. The advantage of JXME, in front of other proposals, is that it is the mobile version of the well known JXTA platform [5]. JXTA is a set of protocols which allow peers to communicate, publish and find resources, and consume remote resources, independently of the actual transport layer and the implementation language. JXME allows mobile devices to create a mobile JXTA network and also to participate in a "wired" standard JXTA network. JXME heavily takes into account the idiosyncrasies of mobile devices such as power and storage limitations, and for that reason research has focused on these features [6]. However, security is a very important issue that has been often forgotten in JXME research.

The main goal of this paper is to analyze the security mechanisms that JXME provides. Such analysis should allow to determine which will be the minimum security features included in P2P mobile applications developed on top of JXME. We based our study in JXME-Proxyless, one of the two available JXME versions, since it is the most complex one and the one where peers are actually self-organized. The security analysis performed in this paper follows the idea of [7] where a generic JXTA security survey has been presented. Applying the same methodology, security is not analyzed by reviewing basic peer operations in an isolated manner, but taking into account the whole peer life cycle. With this approach, it is possible to identify the available security mechanisms and how they operate.

The paper is organized as follows. Section II provides an overview of the JXME project. Section III presents the security analysis of JXME-Proxyless. Section IV provides a brief comparison between both versions of JXME. Finally, Section V outlines the conclusions and further work.

II. Overview of JXME

Currently, only Java implementations of JXME exist. They are direct offshoots of the generic JXTA specification, thus sharing many characteristics with the desktop version. A detailed explanation of JXTA's generic protocols and services can be found in [8], however, we will briefly outline the most important concepts.

In both cases, JXTA and JXME, the architecture is completely based on the concept of *Peer Groups*, sets of peers with common interests which agree on shared services. Peer Groups are managed by the *Membership Service*, one of JXTA's core services. Once a peer has joined a Peer Group, any resource may be shared with other group members by distributing its associated *Advertisement*, an XML metadata document describing the resource properties and how it may be accessed. Advertisements are located and distributed using the *Discovery Service*. A network resource cannot be accessed without previously recovering its associated Advertisement. Every time an Advertisement is retrieved by a peer, it is stored in the local cache and assigned an expiration date. At that date, the Advertisement will be automatically flushed. Once a resource has been located, messaging may begin using JXTA *pipes*, abstract endpoints which provide an asynchronous unidirectional communication channel.

Therefore, the Java implementation of JXME can be viewed as a JXTA compatible platform for resource constrained de-

vices, based on the framework specifications for Java ME: *Connected Device Configuration* (CDC) and *Connected Limited Device Configuration* (CLDC). The CDC specification uses the C-Virtual Machine (CVM), an optimized version of the Java Virtual Machine (JVM) [9], it contains some of the standard Java packages, and it is addressed towards high end mobile devices, such as powerful PDA's and smart phones. In contrast, the CLDC specification uses the Kilobyte Virtual Machine (KVM) [10], which has few of the standard Java packages, thus being suitable for lower end devices with very slow processors and very reduced memory. CLDC is further divided into two profiles which define its operation mode: *Mobile Information Device Profile* (MIDP) and *DOcomo JAva* (DOJA). The former is a specification for the usage of Java on embedded devices and the latter is a Java environment specification for DoCoMo's i-mode mobile phone.

Using JXME, any CDC/CLDC device can participate in the JXTA network and exchange messages with any other peer. Unfortunately, because of the limited capabilities of mobile devices, they cannot fulfill some of the JXTA peer basic functions such as encoding JXTA messages in XML, maintaining a local copy of the network state and listening to incoming network information at socket or datagram level. Two distinct versions of JXME currently exist, each one suitable for a different set of mobile devices. On one hand, the *JXME-Proxied* version is a very simple implementation for limited devices, which delegates all the heavy work to an external super-peer, the *Relay Peer*. On the other hand, the *JXME-Proxyless* version is a more complex one, where mobile peers may directly interact with the JXTA network.

In this paper we focus in the JXME-Proxyless version. We consider it is the most interesting one, since it is the most complex and the one where peers are actually self-organized. However, we will provide insights on JXME-Proxied in Section IV.

A. JXME-Proxyless version

The JXME-Proxyless version currently holds the newest and most complete implementation of JXME, having been expected by the community for years. This version is the nearest one to the JXTA specification, allowing mobile devices to directly participate into the JXTA network by themselves, without the need of an external super-peer. Figure 1 shows the JXME network architecture and how it interoperates with a desktop JXTA network. However, the most advanced functionalities of desktop JXTA, such as the Shared Resource Distributed Index (SRDI) [11], have been implemented as lighter versions, taking into account the limited capabilities of mobile devices.

Any peer using JXME-Proxyless is named a *Proxyless Peer* and is able to perform the following actions by itself:

- Discover other devices and services.
- Publish Advertisements about it's own resources.
- Establish direct connections to any other peer.
- Create/join private virtual domains (Peer Groups).

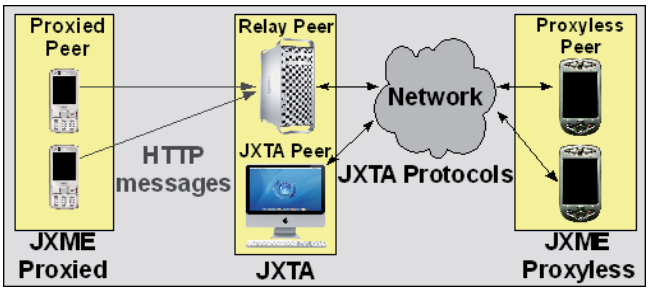


Fig. 1. JXTA and JXME network architecture

- Directly exchange/access content with other Peer Group members.

Proxyless Peers may use JXTA's most important components: Peer Groups, Advertisements and Pipes. Since Proxyless Peers may directly interact with other peer group members under a secure context, Peer Groups are necessary to maintain JXTA's architecture. Advertisements are encoded using XML, just like in the desktop version, in order to maintain language independence. Finally, in spite of its complexity, Pipes are available since direct TCP communications are supported.

However, Proxyless Peers have some limitations on regards to the JXTA base architecture. First of all, even though most JXTA services are implemented, some of they are not, and even when a particular service exists, it must be taken into account that it may not have full capabilities. An obvious example is the Membership Service, which does not support all implementations available in desktop JXTA. Furthermore, another constraint is that a Proxyless Peer cannot act as a super-peer in a JXTA network. As a result, since super-peers help network management, such as caching Advertisements to accelerate search queries, a JXTA network formed only by Proxyless Peers may have scalability issues.

Finally, it is also important to point out that, since Proxyless Peers directly participate in the JXTA network (forwarding messages, finding routes, saving Advertisements, etc.), they have to pay a cost in resource consumption, such as battery power, even when the mobile device is in standby mode.

B. Related work

Some research exists on JXTA, enhancing its basic features [11] and security [7], but not many efforts have been made for JXME specifically.

In [12] an analysis about JXME functionality is found, concluding that JXME-Proxied is not suitable for MANET environments because of its centralized architecture but JXME-Proxyless can fit well in this type of environments. A framework for mobile devices optimized to MANET networks which is compatible with JXTA protocols is developed in [13]. In [14] a framework to allow JXME devices to use bluetooth is presented. This framework permit devices to overcome Bluetooth limitations, such as the maximum number of interconnectable devices and the maximum transmission range.

JXME has also been analyzed and used to build a set of applications. For instance, in [15] JXME-Proxyless is used to implement a distributed collaborative platform which makes people in distributed spaces ubiquitous collaborate with friends and colleagues.

However, regarding security, to our best knowledge, there is no attempts to identify nor improve the JXME-Proxyless security properties.

III. JXME-Proxyless security assessment

Guaranteeing a minimum security level should be one of the main goals in most of the current P2P applications, even though this level may differ depending on the particular needs of each application. In this section, a security analysis of JXME-Proxyless is made in order to evaluate the security level currently provided by the platform. This analysis follows the methodology used in [7], where the general peer life cycle is examined rather than isolated peer actions.

The standard JXME-Proxyless general operation cycle can be summarized in the following steps [7]: Platform startup, Peer Group joining, Resource discovery and publication, Message exchange and Disconnection. A brief description of each step follows:

1) **Platform startup**: This is the first action performed by a JXTA Peer and consists in loading the required libraries and initializing the system prior to going online.
2) **Peer Group joining**: At this step, the peer joins a Peer Group, so interaction with other Peer Group members is possible. Peer Group joining is managed by the Membership Service, one of JXTA's core services, which allows peers to claim unique identities within a Peer Group.
3) **Resource discovery and publication**: Encompasses the distribution and location of Advertisements and how to access it. This action is performed via JXTA's Discovery Service.
4) **Message exchange**: This is the most frequent action in Proxyless Peers, consisting of data exchange, usually in order to access resources, such as available services. This exchange can only exist between same group members and is accomplished using JXTA pipes.
5) **Disconnection**: Peer cleanup before exiting the JXTA network. This is the last action a peer performs before going offline.

A. Attacks in P2P networks

In order to perform a security assessment, it is useful to identify and categorize the most common attack types in P2P networks. All attacks can be divided into two distinct groups, according to the degree of involvement of the attacker [16]: passive attacks, where the attacker just monitors peer activity and network traffic, and active attacks, where the attacker purposely interferes with network activity. Each group can be further classified according to the particular action performed by the attacker.

We are interested in the following attacks:

Passive attacks:

- *Eavesdropping* (Evs): Searching, in message exchanges, for sensitive information such as passwords.
- *Traffic analysis* (TAn): Analyze traffic data, looking for patterns and relevant peers.

Active attacks:

- *Spoofing* (Spf): Impersonating another peer.
- *Man-in-the-middle* (MitM): Intercepting the communications between two parties, transparently relaying forged messages to each one.
- *Playback* (Pb) or *Replay* (Rp): Capturing messages so they can be reused at a later time, simulating a real message exchange initialization.
- *Local data alteration* (LDA): Modifying local data to corrupt the system behavior.
- *Software Security Flaws* (SSF): Exploiting vulnerabilities due to bugs in the source code trying unexpected actions on the software.

B. JXME-Proxyless Security evaluation

From the peer operation cycle and the identification of possible attacks, it is possible to provide a structured security assessment. To identify which vulnerabilities exist, we have designed and performed some attacks which try to subvert JXME operations. Our analysis is focused in active attacks, since they need technical knowledge about the JXME architecture, and rely on active operations to exploit vulnerabilities. Passive attacks are more generic and can be performed using common tools, such as sniffers [17].

1) Platform startup: The first action a Proxyless Peer performs consists in loading the JXTA libraries and creating the default network manager. This operation does not perform any network activity, and thus is protected from external interference at this level. The only vulnerabilities that exist are those related to library authenticity. Since no mechanisms are provided to differentiate a good JXME-Proxyless distribution from a malicious one, it is possible to subvert the system via *local data alteration* attacks.

To prove this flaw, we have designed an attack where an original Proxyless Peer (P_1) tries to send messages to a hacked Proxyless Peer (P_2), who uses a modified JXME-Proxyless library. The attack consists on removing the content of the *publish* and *remotePublish* methods inside the *net.jxta.impl.discovery.DiscoveryServiceImpl* class. Both methods are used to publish and propagate Advertisements to other peers. Therefore P_2 is not able to distribute his Advertisements over the JXTA network. These changes make P_2 unreachable from P_1 and from the JXTA network, since its Peer Advertisement, needed by P_1 or any other peer to route messages to him, is never published.

2) Peer Group joining: The step of joining a Peer Group is handled via the JXTA Membership Service. This is one of JXTA's core services, which manages the group members' identities within the group context. Identities are assigned by successfully completing an authentication process prior

to actually joining the group. The Membership Service is defined as generic in the JXTA specification, leaving up to developers to implement their own version, with the security level required by their applications.

Even though JXTA provides some reference implementations for the Membership Service, JXME-Proxyless provides none at all, allowing any Proxyless Peer to create and join any Peer Groups. Since no Membership Service is implemented, no security really exists for the join operation, and no authentication process is enforced, allowing any peer to claim any identity within the system. We tested this security flaw by running two Proxyless Peers that execute a demo chat application provided with the JXME-Proxyless library. Both peers exchange messages inside a newly created Peer Group. Additionally, we have also created an additional peer who can join this new group, but claiming any identity. This peer is able to send messages to other group members.

3) Resource discovery and publication: In JXTA and JXME-Proxyless, resources are published across the JXTA network by distributing an Advertisement. The JXTA specification defines Advertisement security at two distinct levels: at Advertisement level and during its transport. In the former, the secure layer data is directly included in the Advertisement as additional content, whereas in the latter, the Advertisement is processed as a simple message. Security at message layer will be discussed in Section III-B4.

As far as Advertisement level security is concerned, JXME does not provide any security mechanism. Advertisements are transmitted without any kind of privacy over the network, thus becoming vulnerable to *eavesdropping* attacks, as well as *traffic analysis*, since an attacker can identify important Proxyless Peers (those with many resources) by the amount of published Advertisements.

Furthermore, we have designed and performed a *Spoofing* attack on Advertisement exchanges, where there are two Proxyless Peers (P_1 and P_2) exchanging messages, and a malicious Proxyless Peer (P_3) trying to impersonate P_2. The structure of this attack is shown in Figure 2 and follows the steps:

1) P_2 publishes his Peer Advertisement, containing his route address.
2) Since no mechanism is provided to authenticate peers, P_3 can publish a Peer Advertisement using P_2's identifier but including P_3's address. Once this Peer Advertisement is propagated across the network, it will replace the original P_2 Peer Advertisement.
3) Before P_1 can send a message to P_2, it has to ask for P_2's Peer Advertisement to the super-peer.
4) A super-peer sends P_2's Peer Advertisement to P_1.
5) P_1 tries to send a message to P_2, but he will actually send it to P_3 instead. This attack will be reverted when P_2 republishes his Peer Advertisement.

Moreover, a vulnerability inherited from JXTA still exists. In JXTA, super-peers are responsible for the propagation of Advertisements over the JXTA network. However no mechanisms to identify false ones exist. Therefore, a malicious

Fig. 2. Spoofing attack

super-peer can perform a *Man-in-the-middle* attack between Proxyless Peers inside different networks, and modify any information in the Advertisements prior to propagating them over the network.

As far as *local data alteration* is concerned, since Proxyless Peers' local cache is stored in RAM, and no persistent copy ever exists, they are protected on the long term.

4) Message exchange: This is the most common operation in Proxyless Peers, and therefore, where more efforts have been made by developers to implement security mechanisms. Proxyless Peers exchange messages using pipes, unidirectional and virtual connections between abstract endpoints. JXME-Proxyless supports two different pipe types: Unicast and Propagate. Both are considered unreliable, the former being used for point-to-point communication whereas the latter for one-to-many message broadcast.

Unfortunately, both pipe types have some security issues:

- All data is sent in clear, and thus vulnerable to *eavesdropping*.
- Any data sent through a pipe may actually hop across other peers before reaching the intended destination, which makes the transmission prone to *man-in-the-middle* attacks.
- There is no assurance that a pipe is connected to the specified peer (*Spoofing*).

Fortunately, developers are currently working in the implementation of a wire transport level security layer that may be applied to pipes. This security layer is based in JXTA's own definition of *Transport Layer Security* (TLS) [18]. This definition is based on two distinct protocols:

- Handshake Protocol: Initial TLS negotiation protocol, responsible of the authentication between both peers.
- Record Protocol: Provides a private and reliable communication channel by encrypting data using symmetric cryptography and using integrity check.

This implementation basically allows private, mutually authenticated and reliable communication, protected against both passive and active attacks. Pipes based on TLS are named UnicastSecure, greatly improving JXME-Proxyless security.

However, after performing several tests, we have realized that UnicastSecure pipes provide a secure communication channel but remain unreliable. It is possible to receive an acknowledgement from your messenger without the message actually having been sent. This is because they have been built using the *NonBlockingOutputPipe* java class. Moreover, to perform secure communications using UnicastSecure pipes, an external certificate authority (CA) responsible to manage certificates is required.

Furthermore, unfortunately, UnicastSecure pipes are restricted to point-to-point communications, and therefore cannot be used for message broadcast, which is quite common in a Peer Group context. In addition, no traffic masquerading mechanism is implemented, so it is still open to *traffic analysis*. Finally, the classes needed to implement TLS are not included in the common libraries provided by Java ME. The *Foundation Profile* is required, an optional package which is a standard Java specification and is defined by the Java Community Process (JCP) in JSR 219 [19].

5) Disconnection: Since this operation does not require any communication using the network, no security vulnerabilities exist. This step in JXME-Proxyless is included just for the sake of completeness.

C. Evaluation summary

Even though no software is fully free from bugs, it can be considered that JXME-Proxyless has a big advantage because of its Open Source Software (OSS) nature [20]. Being supported by a community of enthusiastic developers, it can be considered relatively safe from *Software security flaws* on regards to security.

The analysis of possible attacks and the existing security mechanisms of JXME-Proxyless, classified by peer operations, provides a vulnerability map summarized in Table I. Attacks are those described in Section III-A, indexed by abbreviation.

From our experiments, it can be concluded that JXME-Proxyless is vulnerable to the following kinds of attacks:

- **V(1)**: Malicious executable code can easily be built and cannot be automatically discovered when installed.
- **V(2)**: No encryption mechanism exists. Advertisements are transmitted in plain text.
- **V(3)**: No data flow masquerading mechanism exits. It is easy to identify important peers by its traffic.
- **V(4)**: No repudiation method exists. Any peer can publish Advertisements in name of any peer.
- **V(5)**: No repudiation or encryption method exists. Any peer can modify Advertisements.

The available security mechanisms are:

- **P(TLS)**: Transport Layer Security

IV. Brief comparison between JXME-Proxyless and JXME-Proxied

Even though we have focused on the JXME-Proxyless version, in this section we highlight the main differences with the JXME-Proxied security model. Such differences are mainly due to JXTA Proxied's much more minimalist design, which relies on a centralized approach. Any peer using JXME-Proxied is named a *Proxied Peer* and since they are assumed to have very limited resources, cannot directly communicate with other peers within the JXTA network. All messages are exchanged through a *Relay Peer*, a special kind of super-peer which implements the Relay and Proxy JXTA services.

The communication between Proxied and Relay Peers is performed with a simplified protocol based on HTTP. This protocol is performed exchanging text plain messages which contain the operation to execute. The available operations are predefined: **Join** a group, **Search** or **Create** resources (such as Peer Groups or pipes), **Listen** to a pipe to receive data, **Send** data to a specific pipe, **Close** a pipe and **Poll** the Relay Peer for incoming messages from the JXTA network. Basically, it means that Proxied Peers delegate JXTA communications to the Relay Peer and only execute the previous mentioned operations.

The most important difference between both JXME versions is that a Proxied Peer needs a Relay Peer to participate in the JXTA network. Therefore, during the platform startup operation, a Proxied Peer, in addition to loading the required libraries, needs to connect to any available Relay Peer. In this initial communication, the Relay Peer creates the PeerId for the Proxied Peer and sends it in plan text. Since Relays Peers are only able to identify Proxied Peers by their PeerIds, interception becomes a security vulnerability.

In JXME-Proxied, unlike Proxyless, Proxied Peers can only join to Peer Groups which implement the *None Membership Service*, the default Membership Service in JXTA. It is designed for applications with no security concerns, being used in groups without authentication, where any peer can claim any identity. There is an initial implementation of a Membership Service based on passwords, the *Passwd Membership Service*. However, as mentioned in the JXTA documentation, it was designed only for testing, since passwords are still transmitted in clear text across the network. Therefore, we can consider that no effective security is implemented in JXME-Proxied at the join operation.

As far as Advertisement publication is concerned, in contrast to JXME-Proxyless, where they are encoded in XML, JXME-Proxied exchange plain text messages, because in limited devices a XML parser is not feasible. But in terms of security, no security layer over Advertisements is provided either. Therefore, both versions share the same vulnerabilities.

Another important difference in JXME-Proxied exists in the message exchange step. In JXME-Proxied it is only possible to perform outbound HTTP connections, in contrast with JXME-Proxyless where direct input and output TCP connections are allowed. That's one of main the reasons why Relay Peers are required. This approach tries to mitigate resource consumption, since Proxied Peers not being directly connected to the JXTA network, they do not have to forward messages, find routes or save Advertisements. However, in terms of security, while Proxyless Peers can directly send the data in a secure way, Proxied Peers send it in simple text, becoming

Operation/Threat	Evs	TAn	Spf	MitM	Rp	LDA	SFF
Startup	N/A	N/A	N/A	N/A	N/A	V(1)	P(OSS)
*Join**	N/A	N/A	N/A	N/A	N/A	N/A	P(OSS)
Publish/Discover	V(2)	V(3)	V(4)	V(5)	N/A	N/A	P(OSS)
Messaging	P(TLS**)	V(3)	P(TLS**)	P(TLS**)	P(TLS**)	N/A	P(OSS)
Disconnect	N/A	N/A	N/A	N/A	N/A	N/A	P(OSS)

TABLE I
JXME-PROXYLESS PEER OPERATION CYCLE SECURITY SUMMARY
(N/A: NON-APPLICABLE. V(TYPE): VULNERABILITY EXISTS. P(MECHANISM): SECURITY MECHANISM USED)
(*STEP NOT ACTUALLY IMPLEMENTED IN JXME-PROXYLESS)
(**NOT USABLE FOR MESSAGE PROPAGATION)

totally vulnerable to passive and active attacks.

Finally, the Disconnection operation is different in JXME-Proxied, since it is not explicit. A Relay Peer decides when to unsubscribe any Peer or a Peer Group. Proxied Peers can only perform an operation to close a pipe by knowing its id. However, it means that a Proxied Peer is vulnerable to spoofing, even when it is disconnected, until the Relay Peer decides to actually unsubscribe it.

V. CONCLUSIONS AND FURTHER WORK

Even though JXME-Proxyless is supposed to be a version conceptually very similar to desktop JXTA, with lightweight versions of the original core services, its security capabilities are still at its infancy. Only secure pipes have actually been paid attention by the developers. This is one of the evidences that being an OSS project is both boon and bane. On one hand, anyone may audit the code, looking for flaws, and contribute to the project. But on the other hand, implementing actual improvements whole depend on contributors' goodwill or interest.

From the security analysis of the JXME-Proxyless, it can be concluded that, in the current version, developers have started to take into account security, with the inclusion of TLS. Unfortunately, it is important to highlight that only using TLS is not enough to protect the system, since an attacker can easily claim any identity and impersonate any Proxyless Peer during Advertisements publication. Therefore, there's still a lot of work pending. Finally, we can also conclude that the JXME-Proxied version, were priority is in performance and not security, does not have an appropriate security baseline, because messages are exchanged with the Relay Peer in clear text, and no powerful authentication method is provided.

Further research includes providing JXME-Proxyless with an actual Membership Service, to provide authentic peer identities within a Peer Group. Once this service is established, it is possible to protect Advertisements. All these improvements should heavily take into account the idiosyncrasies of mobile devices, in contrast to a desktop environment.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish Ministry of Science and Innovation, the FEDER funds under the grants TSI2007-65406-C03-03 E-AEGIS, CONSOLIDER-INGENIO CSD2007-00004 ARES.

REFERENCES

[1] Skype, "Skype on your mobile", 2004, http://www.skype.com/mobile.
[2] G. Kortuem, "Proem: a middleware platform for mobile peer-to-peer computing", *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 4, pp. 62–64, 2002.
[3] B.G. Christensen, "Experiences developing mobile p2p applications with lightpeers", *Peer-to-Peer Computing, IEEE International Conference on*, vol. 0, pp. 229–230, 2006.
[4] Sun Microsystems, "Project JXME", 2003, https://jxta-jxme.dev.java.net.
[5] Sun Microsystems, "Project JXTA", 2001, http://www.jxta.org.
[6] T. Piedrahita and E. Montoya, "Performance analysis of JXTA/JXME applications in hybrid fixed/mobile environments", *Revista Colombiana De Computación*, vol. 7, no. 1, 2006.
[7] J. Arnedo-Moreno and J. Herrera-Joancomartí, "A survey on security in JXTA applications", *Journal of Systems and Software*, vol. 82, no. 9, pp. 1513 – 1525, 2009.
[8] Sun Microsystems Inc., "JXTA v2.0 protocols specification", 2007, https://jxta-spec.dev.java.net/nonav/JXTAProtocols.html.
[9] T. Lindholm and F. Yellin, *The Java virtual machine specification Second Edition*, Sun Microsystems, 1999.
[10] Sun Microsystems Inc., "J2ME building blocks for mobile devices. white paper on KVM and the connected, limited device configuration (CLDC)", 2000, http://java.sun.com/products/cldc/wp/.
[11] B. Traversat, A. Arora, M. Abdelaziz, M. Duigou, C. Haywood, J.C. Hugly, E. Pouyoul, and B.Yeager, "Project jxta 2.0 super-peer virtual network", Tech. Rep., SunMicrosystems,Inc, May 2003.
[12] G. Paroux, I. Demeure, and D. Baruch, "A survey of middleware for mobile ad hoc networks", in *Technical Report 2007/D004*. 2007, Ecole Nationale Supérieure des Télécommunications.
[13] M. Bisignano, G. Di Modica, and O. Tomarchio, "Jmobipeer: a middleware for mobile peer-to-peer computing in manets", in *Distributed Computing Systems Workshops, 2005. 25th IEEE International Conference on*, June 2005, pp. 785–791.
[14] C. Blundo and E. De Cristofaro, "A bluetooth-based JXME infrastructure", in *Lecture Notes in Computer Science*, 2009, vol. 4803/2009, pp. 667–682.
[15] X. Wang, "Collaboration instance manager of ubicollab 2008", 2008, Master Thesis in Norwegian University of Science and Technology.
[16] D. Brookshier, D. Govoni, N. Krishnan, and J.C. Soto, *JXTA: Java P2P Programming - Chapter 8: JXTA and Security*, 2002, http://java.sun.com/developer/Books/networking/jxta.
[17] G. Combs, "Wireshark", 2006, http://www.wireshark.org/.
[18] The Internet Society, "The Transport Layer Security (TLS) Protocol Version 1.1", 2006, http://www.ietf.org/rfc/rfc4346.txt.
[19] Java Community Process, "Java specification requests (JSR) 219: Foundation profile 1.1", 2003, http://jcp.org/en/jsr/detail?id=219.
[20] J.H. Hoepman and B. Jacobs, "Increased security through open source", *Commun. ACM*, vol. 50, no. 1, pp. 79–83, 2007.

Modelo de procedimiento sancionador electrónico aplicado al control del tráfico vehicular

J. M. de Fuentes, A. I. González-Tablas, A. Ribagorda
Grupo de Seguridad en las T.I.C. Universidad Carlos III de Madrid (España)
Email: {jfuentes,aigonzal,arturo}@inf.uc3m.es

***Resumen*—El incumplimiento de las leyes origina la imposición de sanciones. Una buena gestión de las sanciones se convierte en un factor clave para que éstas sean eficaces. Por este motivo, se han producido impulsos legislativos que persiguen el desarrollo electrónico de los procedimientos. No obstante, hasta el momento no se ha propuesto la realización electrónica del procedimiento sancionador completo en el ámbito del control del tráfico vehicular. En este trabajo se propone un modelo para la implantación del procedimiento sancionador electrónico en dicho contexto, mejorando la capacidad de participación de los ciudadanos interesados en el mismo. Particularmente y por las implicaciones legales del procedimiento, se abordan en detalle los aspectos de seguridad necesarios.**

Palabras clave: **Procedimiento sancionador electrónico, tráfico, sanción, seguridad.**

I. Introducción

A través de la legislación, los estados determinan qué actuaciones están permitidas dentro de su territorio. El incumplimiento de las leyes lleva consigo la imposición de una sanción, que debe ser proporcionada y justa. Para garantizar la observancia de estos principios, se ha definido un procedimiento específico para el establecimiento de las sanciones.

El procedimiento sancionador abarca todo el ciclo de gestión de la sanción, desde la observación del mal cometido hasta que se establece definitivamente la sanción correspondiente. Actualmente, dicho ciclo lleva asociada una notable carga burocrática. Esto origina una gran cantidad de documentación y, al mismo tiempo, la dilatación en el tiempo de los procedimientos. Ambas cuestiones han desembocado en una gestión ineficiente de las sanciones, provocando incluso que haya procedimientos que caduquen sin haberse establecido una sanción.

Los impulsos recientes para la agilización de la Administración Pública tienen por objetivo paliar estos defectos en la ejecución de los procedimientos. En España, esta voluntad se materializó en la Ley 11/2007, de acceso electrónico de los ciudadanos a las Administraciones Públicas [1]. Dicha Ley especifica, entre otras cuestiones, las directrices para la gestión electrónica de procedimientos administrativos.

Uno de los ámbitos donde se gestionan un mayor número de procedimientos administrativos es el control del tráfico vehicular (4,7 millones de procedimientos en 2008[1]). A la vista del elevado volumen de tramitación, la adopción de medios electrónicos (con sus beneficios previstos de transparencia, eficacia y eficiencia) se hace especialmente necesaria. Hasta el momento, sin embargo, no existen propuestas que aborden de manera electrónica todas las fases del procedimiento. Además, las contribuciones que abordan parcialmente este proceso no explotan las ventajas que brinda el desarrollo actual de las tecnologías de la información y que permitiría la interacción en tiempo real con el sancionado y los demás interesados en el procedimiento.

El objetivo de este trabajo es proponer un nuevo modelo de procedimiento sancionador electrónico aplicado al control del tráfico. Dicho modelo abarca todas las fases del proceso, promoviendo la participación directa de los interesados en ellas. Además, el modelo respeta las directrices establecidas por la citada Ley 11/2007, prestando especial atención a los aspectos de seguridad.

I-A. Organización del trabajo

La Sección II describe el proceso sancionador y presenta las normas específicas de gestión electrónica de procedimientos según la Ley 11/2007, con especial atención a las cuestiones de seguridad que ésta plantea. En la Sección III se presentan los principales antecedentes para la realización electrónica del procedimiento. La Sección IV describe el modelo propuesto, analizando sus aspectos de seguridad y presentando un análisis preliminar de la viabilidad técnica del mismo. La Sección V presenta los trabajos relacionados y, finalmente, la Sección VI recoge las principales conclusiones y líneas de trabajo futuro.

II. El procedimiento sancionador en España. Realización electrónica

El objetivo de este trabajo es mejorar el procedimiento por el cual se establecen sanciones administrativas en un ámbito concreto (el control del tráfico). En esta Sección se describe dicho procedimiento y las consideraciones de seguridad previstas en la Ley 11/2007.

II-A. Descripción del procedimiento sancionador

El procedimiento sancionador queda originalmente definido en España en la Ley 30/1992 [2]. Sin embargo, la Ley 11/2007 supuso una renovación significativa del mismo, al proponer su realización electrónica a través de las tecnologías de la información. De hecho, dicha Ley reconoce el derecho de los ciudadanos a relacionarse con las Administraciones Públicas por medios electrónicos [1]. Con ello, se pretende mejorar su transparencia, su eficacia y su accesibilidad para los ciudadanos.

[1] http://www.dgt.es/portal/es/seguridad_vial/estadistica

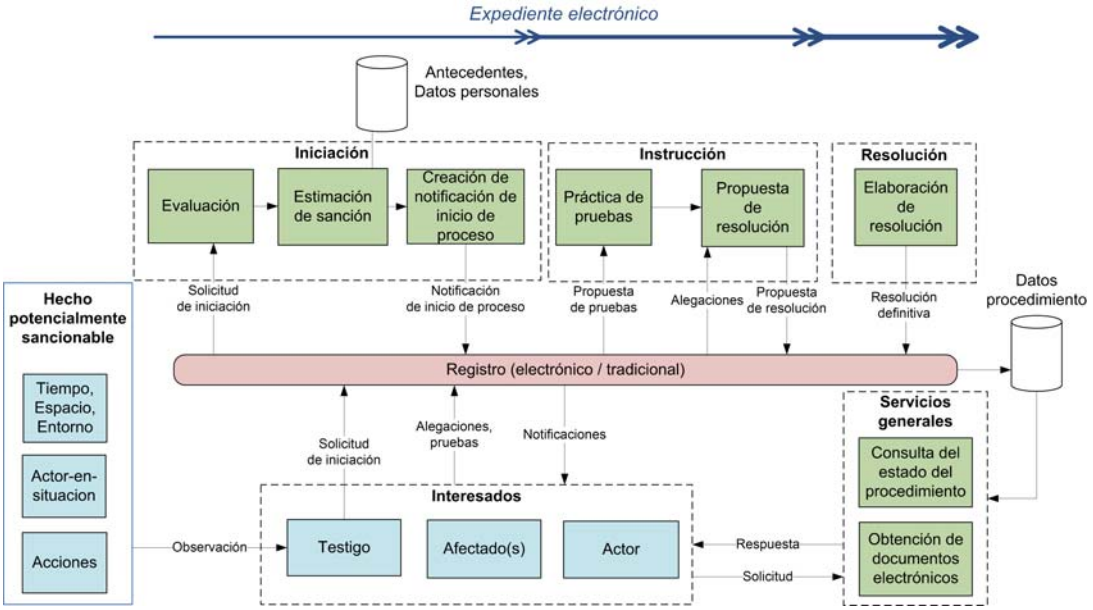


Figura 1. Procedimiento sancionador actual

La Figura 1 describe los diferentes elementos que intervienen en el procedimiento sancionador, incluyendo los medios electrónicos que la citada Ley 11/2007 contempla para su realización. Dicho procedimiento se compone de tres fases (Figura 1, arriba): iniciación, instrucción y resolución.

La fase de *iniciación* puede ser comenzada por cualquier testigo del hecho, presentando una solicitud a través de un registro. Esta presentación puede realizarse de forma electrónica a través de un registro (electrónico), aportando la copia digitalizada de los documentos que en su caso acompañen a la solicitud. Una vez recibida, la solicitud se evalúa, aceptándola o rechazándola en función de los hechos descritos. En el caso de que se acepte, se obtienen los antecedentes y datos personales que serán necesarios para establecer la sanción. Una vez se ha establecido la sanción correspondiente (siempre bajo el supuesto de que los hechos fueran ciertos), se crea la notificación de la iniciación del proceso. Esta notificación se envía a la dirección postal del implicado, salvo que el ciudadano haya permitido el envío a través de medios electrónicos, en cuyo caso así se procede.

Tras la fase de iniciación comienza la de *instrucción*. En ella los interesados pueden formular alegaciones o proponer la práctica de pruebas que permitan adecuar la sanción a la gravedad de los hechos. Así, se puede presentar el testimonio de otras personas presentes en el suceso (alegación) o proponer una pericia técnica sobre el funcionamiento de un dispositivo (prueba). Esta presentación se puede realizar a través del registro electrónico. Tras la valoración de los resultados arrojados por las alegaciones y las pruebas, el organismo instructor efectúa una propuesta de resolución que es nuevamente enviada a los interesados.

Una vez finalizada la instrucción, un organismo distinto del instructor se encarga de la *resolución*, en el que se valora la propuesta de resolución y se redacta la resolución definitiva que pone fin al proceso sancionador.

La tramitación electrónica da lugar a dos nuevos servicios adicionales para el ciudadano relacionados con el procedimiento (Figura 1, derecha). Por un lado, el ciudadano puede consultar electrónicamente los actos de trámite (incluyendo su contenido) y la fecha en que se hicieron. Por otro, el ciudadano puede obtener copias electrónicas de los documentos electrónicos que ya formen parte del procedimiento.

II-B. Necesidades de seguridad para la realización electrónica del procedimiento sancionador según la Ley 11/2007

Para la ejecución electrónica del procedimiento, la Ley contempla una serie de mecanismos técnicos y, sobre éstos, establece ciertas necesidades de seguridad. En este apartado se revisan estas cuestiones, que afectan a dos aspectos distintos: la información del procedimiento y a las relaciones con el ciudadano.

En cuanto a la información del procedimiento, se recoge en el *expediente electrónico*, que contiene *documentos administrativos electrónicos*. Ambos elementos están firmados electrónicamente, a fin de asegurar su integridad e identificar fehacientemente el órgano responsable. Los documentos deben contener, además, una referencia temporal que se presume confiable. Toda esta información se deben almacenar en un *archivo electrónico*, al cual se le exige que proteja la integridad, autenticidad, confidencialidad, calidad, protección y conservación de los documentos guardados. Igualmente, se impone el uso de mecanismos de control de accesos y, en general, que se satisfaga lo dispuesto en la legislación de protección de datos personales.

En cuanto a las relaciones con el ciudadano, se distinguen dos elementos: los registros y la comunicación electrónica. Los registros electrónicos se encargan de recibir y enviar todos

los documentos, escritos y solicitudes que tengan lugar en un determinado procedimiento. Dichos registros deben tener plena disponibilidad. Además, por cada documento aportado por el ciudadano, el registro debe emitir un recibo acreditativo. Dicho recibo consiste en una copia autenticada de lo aportado, incluyendo la fecha y hora de presentación y el número de entrada en el registro.

Finalmente, en la comunicación electrónica con el ciudadano debe quedar constancia de la transmisión y recepción, de sus fechas y del contenido íntegro comunicado. Así mismo, debe identificarse fidedignamente al remitente y al destinatario. Si en esa comunicación se envía una notificación electrónica, debe reflejarse la fecha y hora de la puesta a disposición del interesado de lo notificado, así como la de acceso a su contenido. Los medios empleados para este fin deberán asegurar su disponibilidad, a fin de garantizar el acceso de los ciudadanos a los procedimientos.

III. Antecedentes de tramitación electrónica de sanciones de tráfico con la participación del vehículo

Las redes vehiculares, habitualmente referidas como VANET (del inglés *Vehicular Ad-hoc NETwork*) son un tipo específico de red móvil de comunicación [3]. A través de estas redes los vehículos intercambian datos entre sí y con sistemas externos. Dichos intercambios de información permiten construir nuevos servicios electrónicos, denominados *Sistemas Inteligentes de Transporte* (SIT). Éstos se definen como "*aplicaciones avanzadas que, sin incluir la inteligencia como tal, proporcionan servicios innovadores en los modos de transporte y la gestión del tráfico y permiten a los distintos usuarios estar mejor informados y hacer un uso más seguro, más coordinado y "más inteligente"de las redes de transporte*" [4]. El procedimiento sancionador electrónico presenta dos importantes similitudes con la definición anterior: ambas cuestiones persiguen promover un uso más seguro de las carreteras y, además, buscan ofrecer una mejor información al conductor. En el caso del procedimiento sancionador, la información se refiere al estado de tramitación del procedimiento, cuestión que hasta el momento no ha sido resuelta de forma eficaz en el ámbito vehicular.

El creciente desarrollo de los citados SIT ha dado origen a una arquitectura europea de referencia para estos servicios [5]. Ésta pretende ser el marco común en el que se puedan integrar todos los SIT que se diseñen en un lugar de Europa, asegurando que se pueda incorporar en cualquier otro país de la Unión. Uno de los servicios que se aborda en esta arquitectura es, precisamente, el procedimiento sancionador electrónico. Sin embargo, la funcionalidad prevista sobre este procedimiento se limita a la fase de iniciación. De hecho, la notificación de inicio de procedimiento no se envía a través de medios electrónicos. Dado que se aborda sólo una parte del procedimiento, se disminuye la eficacia y eficiencia esperables del uso de las tecnologías de la información. Por tanto, todavía es necesario proponer contribuciones que permitan realizar el procedimiento sancionador electrónico de forma completa.

IV. Propuesta de aplicación del procedimiento sancionador electrónico al ámbito del control del tráfico

El contexto vehicular está sufriendo una profunda transformación, incorporando las tecnologías de la información en su funcionamiento rutinario. Así, el vehículo y el conductor pueden intercambiar información desde y hacia el exterior. En este trabajo se propone un modelo que abarque todas las fases del proceso sancionador aprovechando el desarrollo de las tecnologías de la información en este contexto. Dicho modelo se basa en el procedimiento sancionador descrito en la Sección II y extiende la funcionalidad prevista en la arquitectura de referencia SIT introducida en la Sección III.

El modelo que se propone tiene dos objetivos principales. El primero es alcanzar una comunicación directa con el interesado, permitiendo que el conductor tenga conocimiento inmediato del estado del procedimiento. El segundo es que el propio interesado pueda crear (y enviar a la Autoridad) pruebas electrónicas que sirvan de base para las alegaciones dentro de la fase de instrucción. Con ello se consigue que el interesado tengan una mayor capacidad de intervención en el proceso.

En esta Sección se describe en primer lugar el modelo propuesto. En el segundo apartado se abordan sus necesidades de seguridad. Finalmente, el último apartado presenta un análisis preliminar de las técnicas de interés para desarrollar la arquitectura derivada de este modelo.

IV-A. Descripción del modelo propuesto

La Figura 2 refleja el modelo propuesto expresado con un diagrama de componentes UML que refleja los cuatro subsistemas que se proponen para realizar el procedimiento sancionador electrónico. Cada subsistema aborda el procesamiento que efectúan cada una de las entidades o infraestructuras implicadas en el procedimiento, esto es, el testigo o testigos, la autoridad o autoridades, los interesados y las infraestructuras de comunicación. A continuación se explica cómo interviene cada uno de los componentes en el proceso resaltando las contribuciones que se realizan para llevar a cabo los objetivos planteados.

El proceso comienza con la detección del comportamiento potencialmente sancionable. Para ello, el Registro de datos sensoriales del testigo muestrea el entorno en busca de actuaciones ilegales. Cuando esto se produce, se emplea el componente de Acreditación y envío para que éste dote de legitimidad a los datos recogidos y se los comunique al organismo iniciador. Dicho envío se realiza a través de la Comunicación hacia la Autoridad y es recibido por su Registro electrónico. Gracias al registro, la solicitud pasa a formar parte de un nuevo expediente electrónico, de lo cual se deja constancia en la infraestructura de almacenamiento a través del componente Gestión de datos. El expediente creado se envía al componente de Iniciación, el cual evalúa lo sucedido y, en base a los antecedentes del infractor (si ha sido identificado) o del titular del vehículo (si el infractor real no ha sido identificado), establece una sanción y envía la notificación correspondiente.

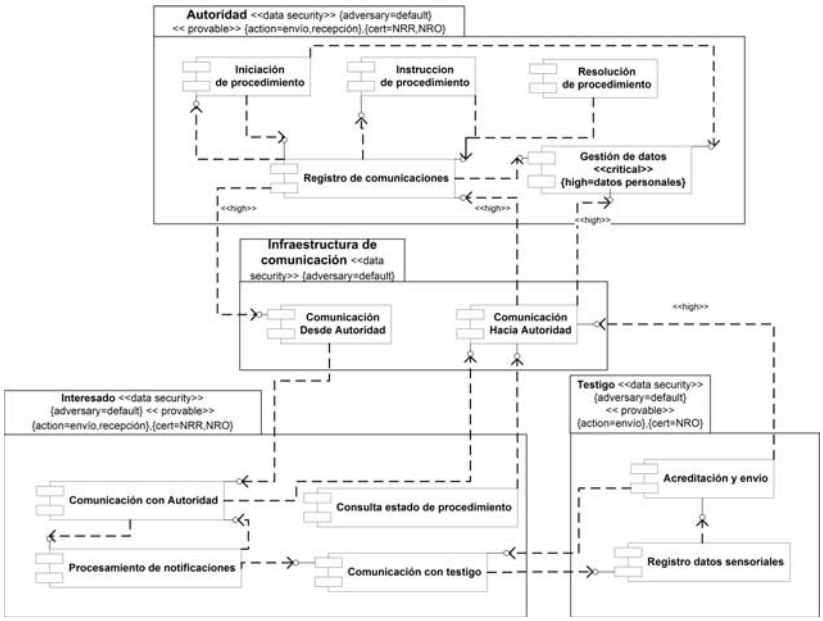


Figura 2. Modelo propuesto de proceso sancionador electrónico en el ámbito del control del tráfico

A diferencia de lo que ocurre en el modelo SIT de referencia (introducido en la Sección III), esta notificación se envía directamente a los interesados (entre ellos, el conductor sancionado). Dicho envío se efectúa a través del registro, quedando reflejado de nuevo en la infraestructura de almacenamiento. Para hacerlo llegar al interesado se emplea el componente de Comunicación desde la Autoridad.

La notificación recibida es interpretada en el componente de Procesamiento de notificaciones, el cual informa al conductor y evalúa la conveniencia de construir las pruebas electrónicas que sustentarán las alegaciones. Los datos que se utilizan para elaborar dichas pruebas proceden de los datos sensoriales (posición, velocidad, dirección, etc.) percibidos por los testigos. Una vez preparada la alegación se envía al componente encargado de la Instrucción del procedimiento a través del componente de Comunicación hacia la Autoridad. Nuevamente, el Registro electrónico se encarga de atestiguar la recepción de las alegaciones y de incorporarlo al expediente electrónico.

El componente de Instrucción del procedimiento valora entonces las alegaciones. En este punto, podría ser necesario realizar pruebas adicionales, como por ejemplo verificar el estado físico del registro de datos sensoriales del testigo iniciador del proceso. En ese caso, el proceso electrónico queda detenido a la espera de la necesaria intervención humana. En caso contrario, a la vista de las alegaciones se establece una propuesta de resolución, que es nuevamente enviada a los interesados del mismo modo que la anterior notificación. El vehículo, nuevamente, interpreta el mensaje recibido e informa al conductor, permitiendo que éste pueda conocer el estado del procedimiento de forma inmediata.

El proceso finaliza con la Resolución del procedimiento. A diferencia de los pasos anteriores, este componente participa en el proceso sin que sea necesario que reciba un mensaje. Esta resolución establece la sanción definitiva, para lo que se debe valorar toda la información contenida en el expediente. En caso de que se hubieran realizado pruebas que interrumpieran la instrucción electrónica del procedimiento, esta fase no podría automatizarse, puesto que la valoración de una prueba tradicional no es computacionalmente alcanzable, al menos a corto plazo. En caso contrario, este procesamiento se efectúa de forma electrónica, enviando a los interesados la notificación resultante. Dicha notificación es finalmente interpretada y su contenido comunicado al conductor.

IV-B. Análisis de seguridad del modelo propuesto

Para que el modelo propuesto tenga cabida en el actual marco legal es necesario satisfacer, al menos, los requisitos de seguridad impuestos por la legislación aplicable (presentados en la Sección II). En esta Sección se describen las necesidades de seguridad que afectan a cada una de las partes del modelo propuesto. Estas necesidades se han incorporado, en la medida de lo posible, en la Figura 2 mediante la extensión UMLSec [6]. Adicionalmente, aquellas que no tenían cabida en dicha extensión se describen en el texto a continuación. Por claridad, se explicarán separadamente los requisitos que afectan a cada uno de los subsistemas que conforman el procedimiento: Autoridad, Infraestructura de comunicación, Interesado y Testigo.

Antes de comenzar la explicación, es necesario caracterizar al adversario frente al que hay que satisfacer las necesidades de seguridad. En el modelo actual se ha escogido el atacante definido por defecto en [6]. Dicho atacante es de tipo externo y puede borrar, leer e introducir mensajes en un canal inseguro, así como borrar datos de un canal cifrado. Este modelo de atacante es razonable para el modelo propuesto, pero debe ser adaptado y extendido en función de la arquitectura que se

derive posteriormente.

IV-B1. Subsistema Autoridad: Este subsistema necesita incorporar la seguridad adecuada para la tramitación del expediente electrónico. Debido a la naturaleza de los datos en juego, es necesario proteger su confidencialidad, lo cual se refleja con el estereotipo *data security*. Por otra parte, se debe evitar que en este subsistema se pueda negar que se envió o recibió una cierta información. Por este motivo, este subsistema tiene el estereotipo *provable*, que se aplica a ambas direcciones del intercambio de información.

En lo que se refiere a necesidades específicas de algunos componentes, el de gestión de datos tiene el estereotipo *critical* dado que alberga todos los datos relativos a los expedientes. Dichos datos son, por su naturaleza, críticos para el procedimiento, por lo que deben extremarse las precauciones en cuanto a su preservacion. Por otra parte, el registro electrónico necesita asegurar su disponibilidad para estar permanentemente accesible.

Finalmente, las dependencias existentes entre este subsistema y los restantes se han marcado con el estereotipo *high*, que refleja que los datos intercambiados son de naturaleza altamente sensible, por lo que deben eliminarse los riesgos de acceso, modificación y borrado no autorizados que puede realizar el atacante considerado.

IV-B2. Subsistema Infraestructura de comunicación: Este subsistema ejerce de intermediario entre el subsistema Autoridad y los demás subsistemas. Se trata por tanto de un subsistema crítico en el conjunto del proceso y por ello se exige su máxima disponibilidad. La transmisión que efectúe este subsistema debe proteger la información en los mismos términos en los que se exige en los demás subsistemas. Por este motivo, se impone a este subsistema el estereotipo *data security*. Debe notarse que éste impone, además, la confidencialidad, integridad, autenticidad y frescura de dichos datos. A diferencia de los demás subsistemas, éste no envía ninguna información por sí mismo, sino que sólo retransmite aquellas que recibe. Por este motivo, en este subsistema no se impone el estereotipo *provable* para la envío de los datos. Igualmente, tampoco se exige para su recepción, puesto que no es el destinatario final de ninguno de los mensajes intercambiados. Sin embargo, sí se exige la auditabilidad de su funcionamiento, a fin de poder verificar (a posteriori) la corrección de su funcionamiento.

IV-B3. Subsistema Interesado: Este subsistema recibe los datos correspondientes al procedimiento y, en su caso, envía las alegaciones a la Autoridad. Dada la sensibilidad de la información en juego, este subsistema se ha marcado con el estereotipo *data security*. Además, debe quedar constancia de la fecha y hora en que se recibe y se procesa la notificación, pues este dato es relevante en el proceso sancionador, tal y como se expuso en la Sección II. Igualmente, no debe poderse negar el envío de las alegaciones. Ambas necesidades se reflejan con el estereotipo *provable* sobre ambos intercambios de información.

Además de lo anterior, sobre este subsistema debe satisfacerse el requisito de disponibilidad, pues puede recibir potencialmente numerosas notificaciones en un corto espacio de tiempo.

IV-B4. Subsistema Testigo: Este subsistema necesita asegurar los datos que gestiona, pues contienen información de carácter personal (por ejemplo, la identidad del supuesto infractor). Esta necesidad justifica el uso del estereotipo *data security*. Por otro lado, el testigo no debe poder negar el envío de la solicitud de iniciación del procedimiento. Esto se refleja mediante el uso del estereotipo *provable*.

La exigida autenticidad de los datos recogidos por el testigo se traduce en dos necesidades interrelacionadas. En primer lugar, los datos sensoriales recogidos deben ser veraces, es decir, reflejar fielmente la realidad. En segundo lugar, dichos datos deben estar referidos al comportamiento de un vehículo convenientemente identificado y autenticado. Esta última necesidad debe satisfacerse sin comprometer la debida privacidad de los conductores. Así, dicho componente, exclusivamente en el caso de que exista infracción, debe obtener el conjunto mínimo de datos para identificar al vehículo y, deseablemente, al conductor. De lo contrario, podría seguirse la trayectoría de un vehículo a lo largo de las carreteras. Dicho seguimiento constituye una violación de la debida privacidad del conductor incluso si solo se identifica al vehículo, en tanto que habitualmente existe una relación entre el vehículo y su conductor. Dicha relación permite que la identificación del vehículo se convierta, indirectamente, en la de su conductor [7].

Además de lo anterior, el testigo debe ser auditable, manteniendo un registro de sus acciones que permita verificar su correcto funcionamiento. Finalmente, el testigo debe estar plenamente disponible, tratando de maximizar la cantidad de observaciones procesadas por unidad de tiempo.

IV-C. Análisis preliminar de viabilidad técnica

El modelo propuesto identifica las principales funciones que permiten abordar de manera electrónica el procedimiento sancionador. Sin embargo, la implementación práctica de dichas funciones queda fuera del alcance del modelo. En esta Sección se introducen algunas tecnologías que pueden ser de interés para abordar esta implementación, satisfaciendo los requisitos de seguridad identificados en la Sección IV-B. Si bien el subsistema Autoridad puede desarrollarse utilizando técnicas de computación más tradicionales, los restantes parecen requerir un enfoque más innovador, pues se implementan en un entorno distribuido donde el vehículo cobra una mayor importancia. A continuación se explican las técnicas que se han identificado en primera instancia como relevantes para el futuro desarrollo del modelo.

Para el desarrollo del subsistema de Infraestructura de comunicación, se pueden aprovechar las diversas tecnologías identificadas por el proyecto CVIS[2]. De entre estas, destacan las alternativas basadas en satélites y las que aprovechan los recientes desarrollos en materia de comunicación vehicular (redes VANET, introducidas en la Sección III). En esta última alternativa, se exige el despliegue de una infraestructura de

[2]http://www.cvisproject.org/

comunicación a lo largo de las carreteras denominada RSU (del inglés *Road-Side Unit*, unidad de comunicaciones longitudinal a la vía). Una de las principales ventajas de su uso es que se permite una comunicación directa y permanente con el vehículo, donde podría alojarse el subsistema Interesado. Para asegurar la conectividad, los vehículos se equiparían con los dispositivos OBU (del inglés *On-Board Unit*, unidad de comunicaciones a bordo). Este tipo de comunicación vehículo-infraestructura dispone de su propia norma acerca de la seguridad de las comunicaciones (IEEE 1609.2, [8]), la cual será de interés a la vista de los requisitos identificados en este trabajo.

Con respecto al subsistema Interesado, se necesita disponer de una plataforma de computación que proteja tanto la información en juego como la corrección de su procesamiento. A este respecto, los dispositivos HSM (del inglés *Hardware Security Module*, módulos de seguridad físicos) satisfacen las necesidades planteadas en el modelo [9].

Finalmente, con respecto al subsistema Testigo, resulta necesario disponer de técnicas confiables de percepción del entorno. Con este fin, los dispositivos EDR (del inglés *Event Data Recorder*, registrador de eventos) permiten registrar lo ocurrido en el entorno vehicular. Además, con el fin de poder incorporar en una visión única las percepciones de varios testigos, se pueden utilizar (o adaptar) protocolos que evalúan la credibilidad de cada uno de ellos [10]. Además, de cara a asegurar los propios sensores y la transmisión de información, pueden aprovecharse los mecanismos propuestos por proyectos de investigación, tales como OVERSEE[3].

V. Trabajos relacionados

Las principales contribuciones técnicas relativas al procedimiento sancionador electrónico en el ámbito del control del tráfico han buscado mejorar la recepción de información por parte del ciudadano. Así, en el caso de España se han implementado las notificaciones telemáticas referidas al tráfico, en las que se envía un mensaje al teléfono móvil advirtiendo de la recepción de una notificación electrónica en la Dirección Electrónica Vial[4]. El modelo propuesto traspasa los límites de esta iniciativa, permitiendo que no sólo se reciba información sino que también se puedan aportar datos al procedimiento.

Por otro lado, el procedimiento sancionador de tráfico se ha modificado recientemente en España [11]. Dicha reforma busca acortar los plazos de tramitación eliminando la fase de instrucción para aquellos conductores que accedan a pagar el importe de la sanción inicial. El modelo propuesto en este trabajo supera a dicha reforma, en tanto que conjuga de una manera razonable los intereses de ambas partes: la Administración desea evitar que los procedimientos se alarguen, pero el ciudadano tiene derecho a defenderse en situación de igualdad probatoria. Para la creación de estas pruebas en el entorno vehicular, existen propuestas previas que pueden servir de base para el desarrollo de la arquitectura derivada del modelo presentado en este trabajo [12].

[3]https://www.oversee-project.com/index.php?id=9

[4]http://www.dgt.es/portal/es/oficina_virtual/multas/notif_por_internet_movil/

VI. Conclusiones y líneas futuras

El procedimiento sancionador permite la imposición de castigos a aquellos ciudadanos cuyo comportamiento no sea acorde con la Ley. Sin embargo, su implementación actual sufre de una excesiva burocracia que redunda en una menor transparencia, eficacia y eficiencia. En este trabajo se ha descrito el procedimiento sancionador y su realización electrónica según la legislación vigente. Partiendo de este marco, se ha propuesto un modelo para su aplicación en el ámbito del control del tráfico con especial atención a los aspectos de seguridad necesarios. Hasta donde se ha desarrollado esta investigación, esta es la primera propuesta que plantea la realización electrónica de todas las fases del procedimiento, lo cual constituye un punto de partida para su tramitación íntegramente automática. El modelo propuesto persigue fundamentalmente incorporar dos características novedosas a las propuestas previas: alcanzar una comunicación directa con el interesado y permitir que éste pueda crear pruebas electrónicas que sirvan de base para las alegaciones. Con ello se persigue proporcionar un conocimiento inmediato del estado del procedimiento a los implicados y otorgarles una mayor capacidad de intervención en el proceso.

Las líneas futuras de trabajo se centran en desarrollar una arquitectura derivada del modelo propuesto, tomando como base las técnicas que se han identificado de forma preliminar en este trabajo.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia e Innovación (España), dentro del Plan Nac. de Investigación Científica, Desarrollo e Innovación Tecnológica 2008-2011, contrato TIN2009-13461 (proy. E-SAVE).

Referencias

[1] España. Ley 11/2007, de acceso electrónico de los ciudadanos a los Servicios Públicos. *Boletín Oficial del Estado*, 23 de junio de 2007, núm. 150, pp. 27150-27166.

[2] España. Ley 30/1992, de Régimen Jurídico de las AA. PP. y del Proc. Admin. Común. *Boletín Oficial del Estado*, 27 de noviembre de 1992, núm. 285, pp. 40300-40319.

[3] L. Le *et al.* CAR-2-X Communication in Europe. En: S. Olariu; M.C. Weigle (eds).*Vehicular Networks: From theory to practice*. CRC, 2009.

[4] Parlamento Europeo. Resolución legislativa por la que se establece el marco para el despliegue de los sistemas de transporte inteligentes en el sector del transporte por carretera. Estrasburgo, 2009.

[5] R. Bossom. *European ITS Framework Architecture, Functional Viewpoint, Version 3*. Proyecto FRAME-S, 2004.

[6] J. Jürjens. *Secure systems development with UML*. Springer-Verlag, 2005.

[7] J. M. de Fuentes; A.I. González-Tablas; A. Ribagorda. "Autenticacion y privacidad en redes vehiculares". En: *Novática*, núm. 202, 2010.

[8] IEEE. *Trial Use Std. for Wireless Access in Vehicular Environments (WAVE) - Security Services for Applications and Management Messages*. 1609.2. IEEE, 2006.

[9] J. Attridge. *"An overview of Hardware Security Modules"*. SANS, 2002.

[10] N. Lo; H. Tsai. "Illusion attack on VANET applications - A message plausibility problem". En: *Globecom Workshops*. IEEE, 2007.

[11] España. Ley 18/2009, por la que se modifica el texto articulado de la Ley sobre Tráfico, Circulación de Vehículos a Motor y Seguridad Vial. *Boletín Oficial del Estado*, 24 de noviembre de 2009, núm. 283, pp. 99594-99624.

[12] J. M. de Fuentes; A.I. González-Tablas; A. Ribagorda. "Witness-based evidence generation in Vehicular Ad-Hoc Networks". En: *Proc. 7th Embedded Security in Cars Conference (ESCAR)*, 2009.

Modelado de amenazas en indexación de páginas y propuesta de inclusión en el ENS

Chema Alonso Cebrián
ESCET
Universidad Rey Juan Carlos
Email: chema@informatica64.com

Antonio Guzmán Sacristán
DATCCCIA
Universidad Rey Juan Carlos
Email: antonio.guzman@urjc.es

Gonzalo Álvarez Marañón
Instituto de Física Aplicada
Consejo Superior de Investigaciones Científicas
Email: gonzalo@iec.csic.es

Enrique Rando González
Departamento de Informática
Delegación de empleo de la Junta de Andalucía
Email: enrique.rando@juntadeandalucia.es

***Resumen*—Este trabajo analiza las amenazas derivadas de las malas prácticas en la gestión de técnicas SEO para indexación de páginas web, así como las vulnerabilidades y ataques que se pueden derivar de ellas. A partir de este análisis se ha propuesto un conjunto de cinco normas que deben resultar básicas para el desarrollo seguro de la gestión de indexación. Además, se ha propuesto la adaptación de estas normas al Esquema Nacional de Seguridad.**

I. Introducción

La correcta indexación de un sitio web por los motores de búsqueda reviste una importancia capital para contar con una presencia sólida en Internet. Con el fin de mejorar el posicionamiento de un sitio web en la página de resultados de un buscador se utilizan las denominadas técnicas de optimización para motores de búsqueda (Search Engine Optimization, SEO). Entre la gran variedad de técnicas SEO, se incluyen la correcta configuración de los archivos robots.txt [1] y de sitemap [2] para indicar a los buscadores qué indexar y qué no dentro de un sitio web. La incorrecta configuración de estos archivos puede acarrear consecuencias negativas desde el punto de vista de la seguridad y del rendimiento de un sitio web.

El viernes 29 de enero de 2010 se publicó en el BOE el Real Decreto 3/2010, de 8 de enero, por el que se regula el Esquema Nacional de Seguridad (ENS) en el ámbito de la Administración Electrónica [3]. El ENS nace con el objetivo de crear las condiciones necesarias de confianza en el uso de los medios electrónicos en las relaciones de los ciudadanos con las Administraciones públicas. Se limita a establecer los principios básicos y requisitos mínimos que permiten una protección adecuada de la información y los servicios [4], en respuesta al Art. 42.2 de la Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los servicios públicos [5].

En el extenso Anexo II, el ENS proporciona medidas de seguridad concretas estructuradas en tres grandes grupos (organizativas, operacionales, de protección), los cuales pueden estar a su vez divididos en más subgrupos. Aunque existe una categoría destinada a la protección de servicios y aplicaciones web, no se tratan específicamente los posibles problemas de seguridad derivados de una deficiente configuración de los archivos robots.txt y sitemaps o una inadecuada gestión del posicionamiento en buscadores.

El objetivo de este trabajo es exponer estos problemas y proponer unas guías de buenas prácticas de cara a combatirlos, las cuales podrían añadirse o complementar las recomendaciones del ENS.

El trabajo está estructurado de la siguiente forma: en la Sec. II se realiza un modelado de amenazas sobre los riesgos derivados de la incorrecta indexación de páginas web; en la Sec. III se ofrecen una serie de recomendaciones para protegerse frente a los riesgos identificados; en la Sec. IV se adaptan estas recomendaciones al formato del ENS; la Sec. V concluye el trabajo.

II. Modelado de amenazas en el contexto de la indexación de páginas web

El modelado de amenazas ayuda a identificar amenazas, ataques, vulnerabilidades y contramedidas con el fin de mejorar la gestión de la seguridad de los sistemas de información. En las siguientes secciones se explican cuáles son las amenazas, vulnerabilidades y ataques a los que está expuesto un sitio web con una incorrecta configuración de los archivos robots.txt y de sitemap.

II-A. Amenazas derivadas de malas prácticas en la gestión de indexación

Se entiende por amenaza el potencial de que un incidente, deliberado o no, comprometa los objetivos de seguridad de la organización [6]. Entre los objetivos de toda organización suelen figurar el salvaguardar la privacidad de la información sensible, así como asegurar un servicio rápido y de calidad. En las siguientes secciones se describe cómo estos objetivos pueden verse amenazados.

II-A1. Revelación de información sensible sobre la organización: Toda organización posee información sensible: datos de personas físicas y jurídicas, ya sean empleados, clientes o proveedores; datos de funcionamiento interno, sistemas y servicios, como archivos de configuración, registros de actividad

y código fuente; etc. Esta información sensible puede revelarse de varias maneras indeseadas y a veces insospechadas.

II-A1a. Metadatos inadecuados en documentos públicos: La mayoría de software utilizado cotidianamente para generar documentos digitales de todo tipo realiza la adición automática de datos sobre los datos creados (metadatos), los cuales se adjuntan de forma más o menos visible a los propios documentos. Estos metadatos pueden revelar información como nombres de personas, organizaciones, fechas de creación, histórico de alteraciones en el documento, rutas de acceso de archivos, dispositivos utilizados en su creación, coordenadas GPS, y un sinfín de datos adicionales.

II-A1b. Errores de sistemas: Todo software está sujeto a errores o condiciones excepcionales que pueden provocar el funcionamiento anormal de una aplicación. Cuando estas excepciones no se gestionan adecuadamente, pueden revelar información sobre el sistema: código fuente, rutas de acceso de archivos, tipo de servidores, versión de software instalado, nombres de usuario, cadenas de conexión a bases de datos, consultas SQL que muestran a su vez estructuras internas de tablas, etc.

II-A1c. Rutas de acceso: Aunque los archivos robots.txt y de sitemap están destinados a los robots de búsqueda, son públicos y cualquiera puede descargarlos. Pueden contener información sobre rutas de acceso, las cuales a su vez revelan qué tipo de software existe instalado y qué contenidos sensibles se desean ocultar.

II-A1d. Contenido de ficheros de configuración: El funcionamiento de algunos servidores se configura mediante archivos de texto, los cuales pueden contener información sensible como nombres de usuario y contraseñas, cadenas de conexión a bases de datos, rutas de acceso de archivos, etc.

II-A1e. Contenido de ficheros de registro de actividad: Registrar en archivos de texto la actividad de un servidor permite estudiar de qué manera es usado y también reconstruir incidencias. Estos registros o logs pueden contener información sensible de los visitantes, como por ejemplo los datos introducidos en formularios.

II-A1f. Evidencias de intrusiones y vulnerabilidades: Hay ocasiones en que un ciberdelincuente consigue vulnerar la seguridad de un sitio web y modificar sus páginas y documentos. El conocimiento público de este tipo de incidentes puede dañar seriamente la imagen y la credibilidad de las organizaciones afectadas. Pero no acaban ahí las consecuencias: si los buscadores indexan y muestran páginas con evidencias de intrusiones o vulnerabilidades, otros ciberdelincuentes podrían aprovechar esta información para dirigir nuevos ataques contra la organización.

II-A2. Deterioro del rendimiento: Un objetivo fundamental de todo servicio web es asegurar un buen rendimiento, percibido por los usuarios como la cantidad de tiempo necesaria para cargar la página solicitada. Los motores de búsqueda legítimos por lo general obedecen el protocolo de exclusión de robots que indica qué porciones del sitio web deben agregarse a los resultados de búsqueda. Archivos robots.txt o sitemaps mal configurados pueden originar una sobrecarga de peticiones por parte de estos robots, causando una pérdida de rendimiento.

II-A3. Deterioro de la calidad de servicio: A medida que se incrementa la complejidad de un sitio web y crece su número de páginas, resulta más difícil navegar por ellas y encontrar la información deseada. Un sitio web que carezca de una buena gestión de SEO perderá visibilidad, ya que no aparecerá entre los 10 primeros puestos en las páginas de resultados de los buscadores, y también calidad, porque aunque aparezca listado, no aparecerán en primer lugar las páginas más relevantes dentro del propio sitio.

En última instancia, una gestión inadecuada de la seguridad y/o de las técnicas SEO pueden causar que el sitio web de la organización sea excluído completamente de las páginas de resultados de los buscadores. Esto puede ocurrir no sólo por una pobre promoción sino también en caso de que las organizaciones que gestionan los buscadores lleguen a la conclusión de que se utilizan técnicas ilegítimas o fraudulentas para mejorar el posicionamiento.

II-A4. Secuestro de resultados de búsqueda: Para asegurar la visibilidad en Internet, es muy importante que la búsqueda de palabras relevantes para el servicio prestado por una organización conduzca al sitio web de esta organización. Existen técnicas conocidas como Black Hat SEO [7] que pueden alterar artificialmente estos resultados.

II-B. Vulnerabilidades en la gestión de indexación: Mala configuración de robots.txt y sitemaps

Se entiende por vulnerabilidad toda debilidad en un sistema que podría permitir o facilitar la materialización de una amenaza contra un activo [6]. La forma de disminuir el riesgo a que se ven expuestos los activos de la organización pasa por mitigar o eliminar las vulnerabilidades. En las siguientes secciones se describen cuáles son las vulnerabilidades más importantes en la gestión de una política de SEO asociadas a los archivos robots.txt y sitemaps.

II-B1. Inexistencia de archivos: Los robots de búsqueda podrían indexar absolutamente todo el contenido al que se tenga acceso públicamente navegando desde la página principal.

II-B2. Archivos excesivamente explícitos: Algunos sitios web se sirven del archivo robots.txt para especificar los directorios o archivos con información sensible para evitar que sean indexados por los robots de búsqueda. Este archivo puede por tanto llegar a contener información sobre directorios y archivos confidenciales.

II-B3. Archivos con errores: Un archivo robots.txt mal configurado puede suponer una sobrecarga para el servidor al permitir que los motores de búsqueda realicen peticiones innecesarias y entren en bucles.

II-B4. Archivos robots.txt y de sitemap mal configurados: Como parte de una estrategia de SEO global, deben configurarse adecuadamente estos archivos para garantizar una buena visibilidad en la página de resultados y una buena calidad en los enlaces mostrados en primer lugar.

```
User-agent: *
Disallow: /administrator/
Disallow: /cache/
Disallow: /components/
Disallow: /editor/
Disallow: /includes/
Disallow: /mambots/
Disallow: /modules/
Disallow: /templates/
Disallow: /installation/
```

Figura 1. Ejemplo de archivo robots.txt con exceso de información.

```
User-Agent: *
Disallow: /etc
Disallow: /bin
Disallow: /tmp
Disallow: /log
Allow: /
```

Figura 2. Ejemplo de archivo robots.txt con exceso de información.

II-B5. Archivos muy permisivos: Permiten que Google indexe todo tipo de páginas de configuración, manuales, ayudas y mensajes de error, los cuales son expuestos a través de búsquedas conocidas como "google dorks" [8].

II-B6. Archivos mal protegidos: Si los archivos robots.txt y/o de sitemap pudieran ser modificados por un ciberdelincuente, éste podría controlar qué contenidos del sitio pueden ser indexados por los buscadores y cuáles no, comprometiendo la imagen pública de la organización y su relevancia y presencia efectiva en Internet.

Por otro lado, si el ciberdelincuente estuviera en condiciones de alterar los documentos publicados en el sitio web, podría conseguir que la organización propietaria del mismo apareciera relacionada con términos o temáticas no apropiadas. También podría introducir elementos que pudieran ser considerados por los buscadores como propios de técnicas SEO ilegítimas o fraudulentas, con las penalizaciones que ello conllevaría para el posicionamiento de la web.

II-C. Ataques

Se entiende por ataque todo intento, exitoso o no, de atentar contra el buen funcionamiento del sistema con el consiguiente incumplimiento de los objetivos de la organización [6]. En las siguientes secciones se describen sin ánimo de exhaustividad algunos de los ataques más populares dirigidos contra sitios web con una pobre gestión de SEO, capaces de materializar las amenazas descritas en la Sec. II-A.

II-C1. Rutas de acceso: El archivo robots.txt de la Fig. 1 contiene un exceso de información al revelar la zona de administración y el tipo de software usado, ya que la carpeta mambots, sumada al resto de carpetas, ofrece el panorama típico de Mambo [9]. Permitió descubrir el software y la ruta de administración para el ataque posterior.

En la Fig. 2 se ofrece otro ejemplo de archivo robots.txt que revela directorios del servidor.

```
Disallow: /educational_games/medicine/
dna_double_helix/xmldata.xml
```

Figura 3. Fragmento de archivo robots.txt.

II-C2. Metadatos: Debido a la mala manipulación de los archivos y a la indexación de Google es posible encontrar información sensible como, por ejemplo, cuentas de usuario con sencillas búsquedas como la siguiente:

```
http://www.google.com/#hl=es&q=
intitle:"Documents and Settings"site:es
ministerio &lr=&aq=f&oq=
intitle:"Documents and Settings"site:es
ministerio
```

Existen herramientas como FOCA [10] que permiten automatizar las tareas de localizar, descargar y analizar los documentos publicados en un sitio web en busca de datos relativos a usuarios, equipos, direcciones de correo electrónico, software y hardware utilizado, etc.

II-C3. Ficheros de configuración: El archivo robots.txt de la Fig. 3 contiene numerosas líneas como la mostrada, en la que se revelan los nombres y rutas de archivos de configuración. En este caso, los archivos ocultados a los buscadores contienen las respuestas al juego propuesto por el sitio web.

II-C4. Revelación interna de datos: En principio, un buscador sólo indexa aquellos documentos a los que se puede acceder navegando a partir de la página inicial del dominio o que figuren en un fichero de sitemap aplicable. Así, si el documento A no está enlazado desde ningún otro documento del sitio web, no será indexado automáticamente por los buscadores. No obstante, si no existe un fichero robots.txt que prohíba la indexación de la ubicación del documento A, un atacante interno podría desvelar este fichero realizando una petición expresa de indexación al buscador con la ubicación exacta del documento A.

II-C5. Alteración no autorizada de archivos: Un atacante que encontrara una vulnerabilidad en un sitio web podría intentar aprovecharla para modificar los contenidos y ficheros de configuración de dicho sitio. De tener éxito, podría ocasionar graves daños tanto a la imagen pública de la organización como a su posicionamiento en Internet. En particular, alterando los ficheros de sitemap y robots.txt podría controlar la forma en que se pide a los buscadores que indexen el sitio, exluyendo documentos relevantes e incluyendo otros con información sensible.

No acaban ahí las posibilidades: algunos ciberdelincuentes trabajan para terceros que quieren mejorar el posicionamiento de sus sitios en los buscadores y, para conseguirlo, insertan enlaces a los mismos en las webs atacadas. Para evitar que estos enlaces sean detectados y eliminados por los webmasters, suelen hacer uso de diversas técnicas que hacen invisibles estos contenidos ante los visitantes humanos, normalmente relacionadas con la ejecución de scripts y el uso de hojas de estilo en cascada.

Pero el uso de estas técnicas es considerado como fraudulen-

to por los buscadores, pudiendo ser causa de penalizaciones en el posicionamiento que, en último término, podrían acarrear la completa exclusión del sitio web de sus páginas de resultados.

II-C6. Indexación de URLs no apropiadas: La mayor parte de los buscadores disponen de servicios que permiten a sus usuarios solicitar la indexación de una página. Un atacante que descubriera una vulnerabilidad explotable a través de una URL podría solicitar la inclusión de la misma en las páginas de resultados, revelando públicamente la existencia de dicha vulnerabilidad. Además, los buscadores ejecutarían el correspondiente ataque cada vez que visitaran la página para actualizar sus índices.

El mismo procedimiento podría ser utilizado para conseguir la indexación de otras URLs no apropiadas.

III. Buenas prácticas en la gestión de indexación

El siguiente apartado recoge algunas de las buenas prácticas que deben ser aplicadas a la hora de exponer un sitio web a las arañas de los buscadores de Internet. Debe hacerse notar que, dado el carácter netamente dinámico del panorama de la Seguridad de la Información, no es posible garantizar un nivel absoluto de seguridad. Por ello, se proponen medidas orientadas tanto a que la información que los buscadores obtengan de la organización sea única y exclusivamente aquella que la organización desea, y su obtención sea efectiva, como a detectar y corregir cualquier impacto no deseado sobre el posicionamiento en buscadores y la presencia en Internet.

III-A. Por omisión: disallow: para todos los robots*

La presencia o no de un sitio web en los buscadores de Internet debe ser una decisión de la organización a tomar en consideración con madurez. ¿Tiene sentido que estén indexados los datos de una aplicación que utilizan sólo los empleados internos de una organización? ¿Tiene sentido que se indexen ficheros y datos privados de aplicaciones en la Intranet? En el caso que desee la organización tener presencia en los buscadores, ¿cómo quiere aparecer en ellos? Éstas y muchas preguntas deben ser contestadas con anterioridad a poner un sitio a disposición de las arañas de los buscadores. Si el sitio ha sido puesto en producción sin haber realizado la reflexión necesaria para conocer la presencia que se desea tener en ellos, debe configurarse un fichero robots.txt que bloquee la indexación de todos los contenidos de la organización.

Debido al gran número de arañas de buscadores, es necesario realizar este bloque para todos los agentes:

```
User-agent: *
Disallow: /
```

Este fichero indica a las arañas que no se desea ser indexado y no volverán a intentar indexar el sitio hasta que, manualmente, se pida su indexación. Si no se realiza esta configuración antes de poner el sitio en producción, los datos de la organización pueden estar copiados durante una cantidad incierta de tiempo en una gran cantidad de buscadores y será necesario realizar un borrado manual en todos ellos.

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /images/
Disallow: /aplicaciones/
```

Figura 4. Ejemplo de un robots.txt para un sitio web.

III-B. Auto-catalogación Sí/No

El siguiente paso consiste en realizar la clasificación que clarifique qué contenido debe o no ser indexado por los buscadores. Hay que tener en cuenta que debe ser indexado aquel contenido que sea estrictamente de índole público. En adelante se entiende por *ruta pública* la ubicación con contenidos que se desean indexar; y por *ruta privada*, la ubicación con contenidos que no se desea que sean copiados a los buscadores.

Para realizar esta catalogación de una forma correcta se recomiendan las siguientes pautas:

- Evitar rutas con contenido mixto (público/privado), ya que provocaría o fugas de información o mala presencia en Internet a la hora de decidir si una ruta es pública o privada.
- Evitar contenido no enlazado en rutas públicas, pues alguien que lo descubra o conozca podrá solicitar su indexación manualmente.
- Evitar rutas privadas conocidas, ya que ubicaciones privadas del tipo /etc o /home pueden identificar la existencia de archivos conocidos, sensibles a la seguridad de la información.
- Evitar rutas privadas explícitas: Una catalogación como privada de una ruta como `/administrator` o `/admin` puede ayudar a un atacante a descubrir la existencia de un fichero login.hml o login.jsp dentro de esas ubicaciones, debido a lo común de estas arquitecturas en aplicaciones web.
- Evitar configuraciones privadas automáticas: ciertas aplicaciones web, como gestores documentales o gestores de contenido, utilizan ficheros robots.txt estándar que son fácilmente reconocidos.
- Evitar el uso de rutas privadas a fichero: El pedir la no indexación de un fichero mediante el fichero robots.txt es hacer pública su ubicación, lo que es igual o más peligroso. Para restringir la indexación de una página única en rutas mixtas existen soluciones tecnológicas creadas para ello como es la meta etiqueta robots:
  `<meta name="robots" content="noindex">`
- Aplicar la misma configuración para todas las arañas de todos los buscadores de Internet.
- Proteger las rutas privadas con listas de control de acceso si es posible para evitar cualquier indexación por parte de los buscadores.

III-C. Optimización del rendimiento y SEO con sitemaps

Para optimizar tanto el consumo de recursos que realizan los robots dentro del sitio como la forma en la que un sitio

aparece en ellos se recomienda hacer un correcto uso de los archivos de sitemap.

Este fichero, aunque es una modificación al estándar original del formato de robots.txt, es de aplicación extendida e indica a los robots de los buscadores tanto la importancia de los ficheros públicos, como su frecuencia de actualización. En sitios en los que se está indexando información estática con pocos cambios se puede configurar un largo periodo de actualización haciendo que el robot no intente indexar nuevamente los elementos. Además, en los sitemaps se marca también la fecha de la última actualización y si ésta es anterior a la fecha de indexación que tiene el buscador, no se volverán a realizar todas las peticiones de documentos.

Usar los sitemaps correctamente ayuda a:

1. Mejorar el rendimiento, aligerando la carga de los bots en el servidor web.
2. Mejorar la presencia del sitio en Internet eligiendo cómo los usuarios deben encontrar y entrar en el sitio.
3. Evitar los ataques de hijacking-SEO [7].

III-D. Auditoría

Una vez terminado de catalogarse correctamente el contenido entre público y privado, y tras optimizarse con sitemaps la carga de los robots y la relevancia del contenido, podría plantearse añadir el sitio a los buscadores mediante la sustitución del archivo robots.txt inicial, que bloqueaba la indexación, por el nuevo archivo generado. Sin embargo, este proceso no debe realizarse hasta que el sitio web haya recibido una auditoría de seguridad, con el fin de que no se indexen posibles páginas de error como consecuencia de vulnerabilidades de Inyección de SQL o de Cross-Site Scripting (XSS).

Además de la auditoría de seguridad, es altamente recomendable realizar un análisis tanto del fichero robots.txt como de sitemap para comprobar que su funcionamiento va a ser el esperado.

Una vez que se haya validado tanto la seguridad del sitio como el formato y la estructura de robots.txt y sitemaps, podrá ponerse en producción.

III-E. Auditoría constante

Debido a la estructura viva de muchos sitios web de Internet, es necesario incluir dentro de los procedimientos de auditoría la revisión de la presencia del sitio en Internet, mediante la reevaluación de robots.txt y sitemaps, como mediante la presencia de posibles fugas de datos en buscadores de Internet para, en caso de haberse producido, solicitar el borrado de la URL de los índices de los buscadores.

IV. Recomendaciones para ENS

Es preciso determinar la forma en que un sistema establece un equilibrio entre la importancia de la información que maneja, los servicios que presta y el esfuerzo de seguridad requerido. Esto supone categorizar el sistema basándose en la valoración del impacto que tendría sobre la organización un incidente que afectara a la seguridad de la información o de los sistemas con repercusión en las funciones de dicha organización.

Para poder mesurar el impacto de un incidente, en el ENS se proponen dimensiones de seguridad sobre las que posteriormente se podrán definir métricas de seguridad. Las dimensiones propuestas son:

a) Disponibilidad (D)
b) Autenticidad (A)
c) Integridad (I)
d) Confidencialidad (C)
e) Trazabilidad (T)

Cada uno de estos aspectos podrá evaluarse con tres posibles valores: BAJO, MEDIO y ALTO, según las definiciones del ENS [3].

Cuando un sistema maneja diferentes informaciones y presta diferentes servicios, el nivel del sistema en cada dimensión será el mayor de los establecidos. De esta forma es posible categorizar un sistema de información en tres categorías: BASICA, MEDIA y ALTA en función de que alguna de sus dimensiones esté evaluada en BAJO, MEDIO y ALTO, respectivamente.

Una vez que se han definido las dimensiones de seguridad relevantes y la categoría del sistema a proteger, es posible elegir qué medidas de seguridad deben implementarse. La selección de las medidas de seguridad implicará la identificación de los tipos de activos presentes y la determinación de las dimensiones relevantes así como de su nivel correspondiente. Estas medidas pueden clasificarse en tres marcos diferenciados: el marco organizativo, el marco operacional y el marco de protección. Este último se centra en la protección de activos concretos, según su naturaleza y la calidad de servicio exigida.

En el Esquema Nacional de Seguridad, a través del anexo II, se propone un sistema tabulado para incluir todos los aspectos que pueden ser estimados como medidas de seguridad. Según se ha visto en las secciones anteriores, surge la necesidad de ampliar la propuesta de medidas de seguridad dentro del marco de protección con un bloque centrado en la protección de las técnicas SEO, en línea con el artículo 42 del ENS, en el que se indica que el esquema se debe mantener actualizado de manera permanente. Se desarrollará y perfeccionará a lo largo del tiempo, en paralelo al progreso de los servicios de Administración electrónica, de la evolución tecnológica y nuevos estándares internacionales de seguridad y auditoría.

En la tabla IV se utilizan las siguientes convenciones:

a) Para indicar que una determinada medida de seguridad se debe aplicar a una o varias dimensiones de seguridad en algún nivel determinado se utiliza “aplica”.
b) « n.a. » significa “no aplica”.
c) Para indicar que las exigencias de un nivel son iguales a las de un nivel anterior se utiliza el signo « = ».
d) Para indicar el incremento de exigencias graduado en función del nivel de la dimensión de seguridad, se utilizan los signos « + » y « ++ ».
e) Para indicar que una medida protege específicamente una cierta dimensión de seguridad, ésta se explicita mediante su inicial.

Dimensiones				Medidas de seguridad	
Afectadas	BAJO	MEDIO	ALTO	mp	Medidas de protección
				mp.seo	Protección de sitios web
C	aplica	=	=	mp.seo.1	Valor por omisión:Disallow para todos los robots
C,D	aplica	=	=	mp.seo.2	Auto-catalogación si/no
D	n.a	aplica	=	mp.seo.3	Optimización del rendimiento y SEO con sitemaps
Categoría	aplica	+	++	mp.seo.4	Auditoría

Cuadro I
CORRESPONDENCIA ENTRE LOS NIVELES DE SEGURIDAD EXIGIDOS EN CADA DIMENSIÓN Y LAS MEDIDAS DE SEGURIDAD

IV-A. Valor por omisión de disallow para todos los robots

Debido al gran número de arañas de buscadores, es necesario realizar este bloque para todos los agentes:

```
User-agent: *
Disallow: /
```

Este fichero indica a las arañas que no se desea ser indexado y no volverán a intentar indexar el sitio hasta que, manualmente, se pida su indexación. Si no se realiza esta configuración antes de poner el sitio en producción, los datos de la organización pueden estar copiados durante una cantidad incierta de tiempo en una gran cantidad de buscadores y será necesario realizar un borrado manual en todos ellos.

IV-B. Autocatalogación: SI/NO

Los sistemas deben decidir qué contenidos son privados y cuales son públicos. A partir de esta clasificación es preciso determinar si las diferentes ubicaciones del servidor corresponden a una ruta pública o a una ruta privada.

IV-C. Optimización rendimiento y SEO con sitemaps

Para asegurar un rendimiento óptimo del consumo de recursos por parte de los robots en un sitio se recomienda una configuración adecuada de los archivos de sitemap. Como resultado se consigue mejorar el rendimiento del sistema, mejorar la calidad de servicio y evitar los ataques de hijacking SEO.

IV-D. Auditoría

Además de las auditorías a las que deberían estar sujetas las aplicaciones informáticas ofertadas por el sitio es preciso realizar un análisis exhaustivo del fichero robots.txt así como de los ficheros de sitemap para validar el comportamiento del sistema y la estructura de estos archivos.

Categoría Básica

Antes de pasar a producción se comprobará el correcto funcionamiento del sistema.

a) Se comprobará que se cumplen los criterios de seguridad
b) Se harán pruebas en un entorno aislado
c) Las pruebas no se harán con datos reales.
d) Se diseñará un sistema de auditoría constante que contemple la naturaleza dinámica de muchos sitios web de Internet y que se traduzca en una reevaluación de las configuraciones de robots y de sitemap. Para ello se deben revisar:
   a) Posibles fugas de datos en buscadores de Internet.
   b) Posibles asociaciones en los buscadores de términos o temáticas no adecuadas al sitio web.
   c) Posibles URLs que revelen la existencia de intrusiones o vulnerabilidades.
   d) Posibles resultados que no se desea que aparezcan en los buscadores.
   e) Solicitar el borrado de URLs no deseadas de los índices de los buscadores en su caso.

Categoría Media

Se realizarán las siguientes inspecciones previas a la entrada en producción:

a) Análisis de vulnerabilidades.
b) Pruebas de intrusión derivadas del uso del sistema de indexación.

Categoría Alta

Se debe contemplar la siguiente línea de actuaciones:

a) Análisis de cumplimiento con la calidad de servicio.
b) Análisis de rendimiento del sistema.

V. CONCLUSIONES

El posicionamiento en buscadores se ha convertido en un activo fundamental para la presencia de las organizaciones en Internet y, como tal, debe ser objeto de una adecuada gestión y protección. En este trabajo se ha propuesto un conjunto de buenas prácticas, adecuando su formulación al Esquema Nacional de Seguridad. En la redacción se ha optado por un enfoque más conciso de lo que es habitual en la propuesta inicial del ENS buscando una mejor aplicabilidad del mismo.

REFERENCIAS

[1] M. Carl Drott, "Indexing aids at corporate websites: the use of robots.txt and META tags", *Information Processing & Management* **38**(2), 209–219, 2002.
[2] http://www.sitemaps.org/es/
[3] Ministerio de la Presidencia, "Real Decreto 3/2010, de 8 de enero, por el que se regula el Esquema Nacional de Seguridad en el ámbito de la Administración Electrónica", BOE 25(I), 8089–8138, 2010.
[4] Antonio Martínez, "Esquema Nacional de Seguridad: Seguridad Obligatoria en las AAPP", *red seguridad* 44, 74–76, 2010.
[5] Jefatura del Estado, "Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los Servicios Públicos", BOE 150, 27150–27166, 2007.
[6] Urs E. Gattiker, *The information security dictionary*, Kluwer Academic Publishers, Boston 2004.
[7] Ross A. Malaga, "Search Engine Optimization–Black and White Hat Approaches", In: Marvin V. Zelkowitz, Editor(s), *Advances in Computers: Improving the Web*, 78, 1–39, Elsevier, 2010.
[8] Johnny Long, *Google Hacking*, Syngress, 2007.
[9] http://www.mamboserver.com
[10] http://www.informatica64.com/foca/

El Paradigma del Agente aplicado en la Ingeniería de Inteligencia Ambiental

Marioli Montenegro, Pablo Antón, Antonio Maña and Antonio Muñoz
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga
Email: {marioli,panton,amg,amunoz}@lcc.uma.es

***Resumen*—El concepto de Inteligencia Ambiental (AmI, Ambient Intelligence) se refiere a un entorno sensitivo, interactivo, interconectado, contextualizado, transparente, inteligente y que actúa sin la intervención humana. Este entorno está emparentado con la ubicuidad de los dispositivos de computación que le permiten, de una forma transparente, advertir los cambios de contexto y reaccionar conforme a ello, tomando la iniciativa de forma inteligente para satisfacer las necesidades humanas dentro de lo posible. La importancia de la seguridad, privacidad y requisitos de confiabilidad, así como la complejidad de estos retos se ve amplificada con el modelo de computación propuesto por la AmI. Desde la perspectiva de la ingeniería del software, el cambio hacia la AmI podría deberse a los mecanismos necesarios para satisfacer los requisitos del cambio del paradigma de objeto hacia el propio paradigma del agente. Los objetos proporcionan funcionalidades que pueden ser utilizadas, en cambio, los agentes saben cuándo y cómo utilizar las suyas de forma autónoma. El paradigma del agente es muy útil para la implementación de AmI, además argumentamos de forma similar que es la base para la ingeniería de estos sistemas, teniendo en cuenta la seguridad desde las primeras etapas en la construcción de los sistemas.**

I. Introducción

Tanto los portátiles, como las PDAs y los teléfonos móviles de tercera generación cuentan con algún tipo de conexión inalámbrica que nos permite el acceso a diferentes redes de datos en cualquier momento y prácticamente en cualquier lugar en que nos encontremos. La evolución en tamaño y capacidades de estos dispositivos y de las comunicaciones inalámbricas han permitido la conexión permanente de los usuarios a la red. Este incremento en la movilidad se puede asemejar al que hubo años atrás desde el clásico ordenador de sobremesa al portátil y va acompañada de una rápida adaptación por parte de los usuarios a este nuevo modelo de comunicación. El siguiente paso sería ayudar a la gente a ser capaces de apreciar la existencia de otras máquinas [1]. Las expectativas para el nuevo software son que será capaz de manejar conceptos como la localización y percepción del contexto, personalización, adaptabilidad, crecimiento orgánico, movilidad y otras muchas características que imponen la necesidad de un software más completo con métodos de ingeniería del software y nuevos e innovadores lenguajes de modelado [3].

La Inteligencia Ambiental (AmI) se centra en adaptar nuestro entorno sensitivo a nuestras necesidades y responder de forma inteligente a los usuarios y a los cambios de contexto del propio entorno [4]. Se espera que los objetos que nos rodean en nuestra vida cotidiana actúen de manera autónoma sin necesidad de la intervención humana. Este tipo de ambientes y la computación ubicua están íntimamente relacionados, ya que se pretende proporcionar a las personas servicios y utilidades de forma transparente, sin que explícitamente lo soliciten. El paradigma del agente es muy prometedor para la implementación de sistemas complejos como los de comercio electrónico, tráfico aéreo, planificación de recursos corporativos, etc. [5]. Inicialmente creado en el seno de la Inteligencia Artificial, la comunidad científica y los investigadores han encontrado otras áreas en las que explotar de forma fructífera el paradigma del agente. Como se defiende en [6], [9] este paradigma ha tenido un especial interés en la ingeniería del software, ya que cambia el paradigma de la orientación de objetos. Este cambio se basa en la apreciación del mundo como una sociedad de elementos inteligentes llamados agentes, que tiene percepción y pueden decidir. Esta manera de ver el mundo se diferencia del de la orientación a objetos, ya que ve conceptualmente al mundo como una colección de objetos sin autonomía.

Una de las debilidades de la AmI es la falta de modelos y prácticas de ingeniería del software que ayuden en el análisis de requisitos, diseño, verificación, testeo, etc. Hasta el momento la investigación en esta área está situada en sus primeras etapas, y la necesidad de metodologías de desarrollo viables está más que reconocida. Nosotros creemos que el paradigma del agente no es útil únicamente para los sistemas de AmI, sino que además en todas las fases del ciclo de desarrollo tiene una gran utilidad. Somos conscientes de las nuevas metodologías de la ingeniería del software orientadas a agentes, aún en dominios académicos, deben ser nuevamente revisadas para la ingeniería de la AmI. Si damos pie al análisis y diseño de tales sistemas por el uso del software dirigido por la ingeniería del software de agentes, tendríamos que obtener sistemas finales basados en agentes que sean robustos, escalables y suficientemente inteligentes para satisfacer las necesidades de los sistemas de AmI.

El resto de este artículo está estructurado de la siguiente forma; la siguiente sección muestra la AmI como un sistema multidisciplinar complejo. La sección 3 muestra una revisión del paradigma de agente. La sección 4 introduce las investigaciones en las metodologías orientadas a agentes. La sección 5 discute el potencial que el paradigma de agente podría tener para el desarrollo de entornos inteligentes robustos. En la sección 6 describimos la importancia de la seguridad desde sus

etapas iniciales y para terminar mostramos unas conclusiones en la sección 7.

II. El aspecto multidisciplinar de la AmI

Estamos enfocando nuestros esfuerzos hacia un modelo de ambiente que es perceptivo, inteligente y activo, que a la vez incluye múltiples disciplinas para contribuir en la creación del escenario final. Un gran número de investigadores están interesados en este nuevo área, evidentemente con diferentes motivaciones y objetivos. En el resto de la presente sección, vamos a proporcionar una visión de la AmI y vamos a intentar capturar una variedad de disciplinas que son indispensables reunir para conseguir esta visión.

Philips [10] tiene una visión de la AmI basada en tres elementos claves, 1) la ubicuidad, referente a los dispositivos interrelacionados con el entorno humano, 2) la transparencia de estos sistemas de computación, es decir, que están ocultos en un segundo plano, 3) y la inteligencia, ya que deben actuar en lugar de estar esperando los mandatos de los humanos.

La visión de la AmI del MIT [11] presenta ciertas similitudes, de hecho, la ve como una integración discreta de la computación en nuestras vidas diarias. Estos medios de computación proporcionan información relevante a los humanos y realizan tareas cuando estos las necesitan. Estos ambientes realizarán las tareas apropiadas de una forma transparente, invisible e inteligente. Tradicionalmente, los ordenadores han trabajado como un mediador o mensajero que aparentemente está entre los humanos y el entorno. En la AmI, esta relación es reemplazada por una relación directa no perjudicial entre los humanos y el entorno en los que estén localizados. En resumen, la computación en la AmI deja de ser visible al usuario.

La primera perspectiva de ordenadores invisibles fue dirigida por Weiser [12]. Esta visión suponía la existencia ubicua de computación y de comunicación en cualquier momento y en cualquier lugar. La AmI se centra en la inteligencia y la conciencia de la ubicuidad de los dispositivos interconectados, de forma que la computación comienza a tomar la iniciativa en nombre de los humanos. La ubicuidad de la computación es la base sobre la que se construyen la AmI, los términos computación ubicua, computación omnipresente, ambientes de computación y AmI son usadas ahora de forma intercambiables con algunas diferencias en el contexto y el énfasis.

Actualmente la AmI está integrando dispositivos en el entorno en el que vivimos, diferenciándose así de la realidad virtual que lleva el mundo a los computadores [12]. Este hecho hace invisibles a los computadores, borrándolos de la conciencia humana. Evidentemente, para llegar a este punto, los ordenadores han tenido que adaptarse a las necesidades y características de los usuarios, en contraste con la escena tradicional en la que se suponía que el usuario se adaptaba a los sistemas de computación. Con la AmI, los artefactos encapsulan de forma implícita el papel de mediador de computación. Los artefactos aparentan tener su propio carácter, autonomía e inteligencia, llegando a ser más agentes que objetos normales.

Como consecuencia de todo esto, la AmI es un paradigma de naturaleza multidisciplinar [13]. La inteligencia distribuida es necesaria para cubrir este entorno inteligente, que ahora está compuesto de unidades de inteligencia distribuida a los que llamamos agentes. Se está diseñando nuevo hardware, necesario para que los dispositivos embebidos sean invisibles y se mimeticen con el entorno físico que los rodea. El sistema de AmI se sitúa en un entorno extremadamente dinámico y abierto a continuos cambios, estos cambios necesitan ser identificados e interpretados para satisfacer las necesidades de los usuarios.

La invisibilidad de los computadores fue considerada por Weiser [12] como una de las más relevantes características tecnológicas. La interacción entre ser humano y ordenador es llevada a un nivel superior, la interacción entre el ser humano y el propio entorno [1]. Las nuevas ideas al diseñar la interacción humano-máquina se trasladan de una interacción explícita a una implícita [14]. La interacción implícita incluye la noción de entrada de datos implícita, conocida comúnmente como contexto [15].

La conciencia del contexto [16] [17] es una característica esencial que los sistemas de AmI han de tratar para actuar de una forma adaptativa e inteligente. Este contexto, que debería ser espacio-temporal, ambiental, personal y social, necesita ser analizado y razonado según [2]. El razonamiento del contexto necesita de un modelo de formalización que actúe como una base de conocimiento y permita inferir en un nivel de conocimiento superior.

De los sistemas de AmI también se espera la habilidad de aprender y de mantener la pista del comportamiento histórico de la persona. De hecho la personalización del software es una de las líneas de investigación actuales con más actividad, sin embargo, deberíamos considerar si un sistema de AmI es útil si se comporta de la misma manera ante diferentes usuarios con diferentes características. La movilidad social de los seres humanos es otro aspecto importante que debe considerar una aplicación de AmI. Normalmente la gente desempeña más de un rol social, estos deberían de estar recogidos por la tecnología considerando su contexto social [18].

La AmI abarca muchos aspectos sociales que deberían ser estudiados y analizados antes de tener una aceptación social en la práctica. La ubicuidad de la computación podría tener cierto impacto negativo. La gente sentirá que pierde el control y que no debe confiar en la tecnología. De hecho, la gente ya ha perdido cierta privacidad con los sistemas de localización de los teléfonos móviles o el uso de las tarjetas de crédito. La computación en AmI se supone que controla varios aspectos de la vida diaria de las personas. Un principio esencial de esta perspectiva es que el humano no sienta que ha perdido el control, y permitirles configurar sus necesidades fácilmente, como podría ser a través de algunos patrones de privacidad. A pesar de todo, existen muchos dominios de interés práctico que pueden terminar beneficiando a los escenarios de AmI, en particular aquellos especializados en el cuidado de personas ancianas y en la ayuda a personas con problemas de demencia, en los que la AmI podría desempeñar el papel de cuidador.

III. El paradigma del agente

La computación basada en agentes está tomando importancia como área de investigación. Este creciente interés está motivado por la necesidad del software de poder actuar sin intervención del usuario y que sea capaz de llevar a cabo el concepto de agencia. No es fácil dar una definición de agente, entre otros factores porque no existe un consenso acerca de las principales características que debería reunir un agente software. Una definición bastante aceptada por la comunidad la podemos encontrar en [7]:

> *an agent is a computer system situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives*

Por su naturaleza un agente tiene el control sobre su propio estado, su comportamiento y es capaz de obtener información del entorno. Estar dentro de un entorno y sentirlo implica la necesidad de que un agente pueda reaccionar a los cambios de contexto del mismo entorno. Entre las características clave que debemos encontrar en un agente encontramos: autonomía, pro-actividad, reactividad, localización, direccionalidad y habilidad social.

La autonomía de un agente viene dada por el hecho de que este se comporte de forma independiente de acuerdo al estado que encapsula. La pro-actividad significa que el agente es capaz de tomar la iniciativa sin órdenes externas. Los agentes tienen objetivos y operan para conseguirlos. Esta característica es bastante más compleja que actuar ante estímulos directos como es habitual en los objetos tradicionales. La localización, del inglés situatedness, se refiere a la habilidad de un agente de situarse en un entorno con más agentes, percibirlos y responder a los cambios que pueda haber. La direccionalidad, del inglés directedness, significa que un agente tiene un objetivo, este objetivo representa el razonamiento de las acciones que realiza el agente. Los agentes viven en sociedades con otros agentes de manera colaborativa o posiblemente competitiva. De hecho, los agentes tienen la habilidad social de interactuar con otros agentes.

Como hemos mencionado anteriormente, un agente está diseñado para convivir en una sociedad de agentes. Un sistema multi-agente (multi-agent system, MAS) es un sistema compuesto por varios agentes que son capaces de alcanzar sus objetivos, difícilmente alcanzables individualmente. Un MAS podría ayudarnos a dividir un problema en componentes que sean capaces de interactuar y tratar con situaciones impredecibles como podría ocurrir en sistemas de AmI.

Un MAS representa una forma natural de descentralización, donde hay agentes autónomos trabajando como nodos con su propio comportamiento y control. Cada uno de estos agentes ve el mundo desde su propia perspectiva y tiene sus propios objetivos e intenciones. Una de las ventajas de los MAS es el buen funcionamiento con sistemas complejos abiertos, y una escalabilidad creciente. Como paradigma de computación es bastante prometedor y nos permite implementar multitud de aplicaciones de diferentes dominios como comercio electrónico, planificación de recursos corporativos, control de tráfico, etc. [5]. Por todo esto, vemos que un sistema de AmI encaja a la perfección con la naturaleza de los sistemas de agentes, tema que abordamos en mayor profundidad en las siguientes secciones.

IV. Ingeniería del software orientada a agentes: Agent Oriented Software Engineering (AOSE)

La ingeniería del software difiere de las demás ingenierías por su fuerte dependencia a las habilidades que tenga el ingeniero a la hora de analizar el problema, diseñar una solución aceptable y realizar el sistema final [19]. A pesar de que, por naturaleza, la ingeniería del software es bastante cualitativa, constantemente se investiga en encontrar métodos científicos, modelos y criterios adecuados que ayuden en el desarrollo de software adecuado para alcanzar los objetivos.

Muchos de los proyectos industriales han fracasado, debido a que el producto software final no realizaba lo que se esperaba de él. Los modelos usados para describir los requisitos y el diseño del software han de ser compactos y expresivos, con el objetivo de poder reemplazar, de manera eficaz, al lenguaje natural. Por lo tanto, los modelos necesitan ser suficientemente precisos, y deben mantener el significado de los conceptos reales que supuestamente representan. Estos modelos han de ser formales, de manera que podamos razonar sobre ellos y descubrir anomalías, inconsistencias o incoherencias. La metodología de la ingeniería del software debe por lo tanto proporcionar un nuevo proceso de creación de este tipo de modelos.

Así como un agente observa constantemente el entorno, lo interpreta, actúa y se comunica con otros agentes, también representa un paradigma de computación muy prometedor para la implementación de sistemas complejos. Las AOSE intentan analizar y diseñar sistemas complejos para llegar a una implementación con agentes. Existen diversos grupos de investigación trabajando en el desarrollo de sus propias AOSE [9]. La orientación a agentes no implica que estas metodologías usen los conceptos de agencias ni las nociones abstractas de agentes en cada una de las fases de desarrollo de software, sino que, el objetivo consiste en analizar y diseñar la manera de obtener sistemas multi-agentes finales. Sólo Tropos [20], como AOSE, utiliza las nociones de agentes y la naturaleza de esta tecnología desde las primeras fases de análisis hasta las últimas, relacionadas con la implementación.

Puesto que el uso de ordenadores se está convirtiendo de manera incremental en una parte esencial de las actividades diarias de los individuos, negocios y organizaciones, y que existe una necesidad de combinar rápidamente entre diferentes nodos y partes de computación, se hace crítico encontrar software dinámico, flexible, adaptable y localizado. De hecho, la necesidad de que el software evolucione crece más rápido que el proceso de desarrollo del software. La AOSE ha comprendido esta necesidad, y está intentando crear métodos que permitan el desarrollo de un software resistente a la evolución de requisitos, un software que sea inteligente para adaptarse y cambiar fácilmente hacia nuevos entornos y requisitos.

Afortunadamente la AOSE no está restringida ni influenciada por ningún paradigma de computación ya creado. Esto si ocurrió con la orientación a objetos y el análisis y diseño estructurado [21] [22]. La disciplina de la AOSE crece al ritmo de la programación orientada a agentes y la infraestructura de agentes, lo cual podría cubrir el hueco existente entre el dominio de los problema y las soluciones.

V. Aprovechando el paradigma del agente en el desarrollo de AmI

El paradigma del agente encaja perfectamente en la implementación de escenarios de AmI debido a las coincidencias entre las características del paradigma y las necesidades de la AmI. La abstracción del paradigma del agente contribuye al desarrollo de sistemas de AmI, incluyendo las fases de análisis y diseño, además de temas de seguridad. Proporcionar seguridad a un ecosistema de AmI lleva consigo satisfacer los requisitos de seguridad de los propietarios de los diferentes elementos como el hardware, el software y la información involucrada en los propios ecosistemas. Uno de los puntos más importantes es la falta de un modelo que pueda describir de manera apropiada este tipo de conjuntos de objetivos de seguridad y fiabilidad. Este hecho, junto con la capacidad de estas propuestas de ser extendidas y usadas en tiempo de ejecución con la ayuda de herramientas automáticas, mejora considerablemente uno de los aspectos más atractivos de la tecnología de agentes para los ecosistemas de AmI. En esta línea, el software de agentes pude ser la base para las nuevas soluciones de seguridad, como encontramos por ejemplo en la protección de contenidos [8] o la distribución de video streaming en multicast [40]. En esta sección vamos a plantear nuestra visión inicial de la orientación de agentes para la ingeniería y seguridad de la AmI.

V-A. Desarrollo de AmI orientada a agentes

Como explicábamos anteriormente, la AmI muestra un grado de complejidad que interrelaciona múltiples disciplinas y que requiere el uso de un paradigma de ingeniería especial. Esta necesidad viene de la nueva naturaleza de estos sistemas, donde el comportamiento no se conoce en detalle. La AmI se distingue por su dinamismo, por ser abiertos y por interrelacionarse con los componentes del entorno. Comparado con las prácticas de ingeniería del software orientada a objetos, el paradigma de agente ofrece un nivel superior de abstracción adecuado para sistemas complejos de ingeniería [23].

Podemos tratar la complejidad del desarrollo de software complejo a través de algunas técnicas como son la 1) Descomposición del problema en subproblemas más pequeños fácilmente manejables. 2) El uso de modelos abstractos para representar sistemas centrados en algunos conceptos y relaciones, omitiendo otros que no están relacionados. Estos modelos deberán ser compactos y suficientemente expresivos para poder resumir de forma útil lo que expresamos en el lenguaje natural. 3) La definición y el manejo de interrelaciones entre componentes para resolver problemas mediante algún tipo de organización jerárquica [25].

Como se indica en [19], el paradigma del agente no es únicamente útil para la construcción del software sino que además se puede usar como una nueva forma de análisis y de diseño de sistemas complejos. Utilizando técnicas de abstracción, descomposición y organización para tratar la complejidad de estos sistemas, se puede seguir el paradigma del agente desde las fases más tempranas. Descomponer un sistema complejo en subsistemas relacionados, cada uno con su propio control de hebras y objetivos, puede verse como una sociedad de agentes que interactúan entre ellos. Los subsistemas se ven como agentes autónomos, la habilidad social del agente implica la interrelación a alto nivel entre los subsistemas. Esta interacción podría modelar la cooperación, coordinación o incluso la negociación entre los agentes. Como para una estructura de organización dinámica, el paradigma de agente tiene la expresividad de representar estos conceptos debido a su estructura explícita y mecanismos flexibles. Existe una metodología conocida con el nombre de Gaia [26] que fue desarrollada para reflejar esta idea proporcionando una forma metodológica para la ingeniería de algunos tipos de sistemas complejos.

Para la ingeniería de AmI, como por ejemplo un campus inteligente, necesitamos descomponer el sistema en subsistemas autónomos, y abstraer usando la conceptualización del nivel de conocimiento a un grano más fino que en la orientación a objetos que resulta útil para comportamientos predecibles y sistemas relativamente estáticos. Con AmI no hablamos de una organización con comportamiento conocido, proceso de negocio y un control directo. Aquí el ambiente está en continuo cambio y de formas impredecibles en muchos casos, de forma que necesitamos un alto grado de adaptabilidad para englobar todas las posibilidades que la AmI va a proporcionar en los escenarios de la vida diaria con multitud de alternativas. Si consideramos la AmI como un sistema complejo abierto, creemos que el paradigma del agente puede contribuir en el análisis y el diseño de escenarios de AmI y no sólo en la implementación.

V-B. Integración de la seguridad en los ecosistemas AmI

Hasta el momento hemos mostrado que los sistemas basados en la tecnología de agentes pueden aportar importantes beneficios, especialmente en escenarios de aplicación distribuidos, autónomos, inteligentes o auto-organizados. Además del alto nivel de autonomía y la capacidad de auto-organización de los sistemas de agentes, estos proporcionan un soporte excelente para el desarrollo de sistemas en los que la fiabilidad es un requisito esencial, ya que encajan perfectamente en el perfil de software de monitorización. Tanto los escenarios de AmI como los de computación ubicua comparten este tipo de características, por lo que los sistemas de agentes son ideales para la aplicación en este tipo de entornos. Sin embargo, a pesar de la atención prestada a los agentes por parte de la comunidad investigadora, esta tecnología no ha conseguido ganar una gran aceptación, especialmente en la industria, y únicamente se ha aplicado en escenarios del mundo real muy concretos y escasos. Este hecho se produce, en gran medida

por la existencia de ciertos problemas de seguridad, que por otra parte son considerados los inconvenientes más urgentes a resolver, en esta tecnología, antes de que los agentes estén listos para ser usados de una forma extendida dentro del panorama industrial. En lo que respecta a la seguridad, los agentes presentan las soluciones más apropiadas puesto que engloban los requisitos de seguridad desde diferentes puntos de vista, con idea de conseguir los objetivos de cada parte de una forma colaborativa. Evidentemente, esta tecnología no estará suficientemente madura para ser usada hasta que se resuelvan los aspectos más importantes en lo que a seguridad respecta, tanto en encontrar los mecanismos apropiados, como en la aplicación de estos mecanismos.

Tradicionalmente se han aplicado algunos de los mecanismos de protección genérica de software para la protección de sistemas de agentes. Sin embargo, las características especiales de esta tecnología obligan al uso de soluciones específicas. Algunos mecanismos de protección están orientados a la protección de los hosts del sistema frente a agentes con carácter malicioso. La solución más relevante para este problema pensamos que viene por la utilización del concepto del Sandbox, consistente en un contenedor que limita o reduce el nivel de acceso que los agentes tienen y proporcionan ciertos mecanismos para controlar la interacción entre ellos. Otra técnica que trata esta vertiente del problema se conoce como proof carrying code (o código portador de prueba) [29], esta técnica se basa en que cada fragmento de código incluye una prueba detallada que se puede usar para determinar si la política de seguridad del host se satisface por parte del agente. Los hosts sólo tienen que verificar que la prueba es correcta (lo que significa que corresponde con el código) y que es compatible con la política de seguridad local, aunque hemos de mencionar que la aplicación de esta técnica no resulta sencilla, de hecho en multitud de ocasiones no es fácil encontrar estas políticas de seguridad.

Otros mecanismos están orientados a la protección de los agentes frente a posibles agencias (hosts) maliciosas. La más representativa es la que se conoce como el uso de santuarios [30], que consisten en entornos de ejecución en los que el agente móvil puede ejecutarse de forma segura. La mayoría de estas propuestas se construyen asumiendo que la plataforma donde se implementa el santuario es segura. Desgraciadamente, asumir esto no funciona en la práctica para los sistemas basados en agentes, puesto que no podemos basarnos en que todas las plataformas sean confiables.

Por otro lado, se pueden aplicar varias técnicas para verificar la propia integridad. Esta idea resulta de utilidad para evitar que tanto el código como los datos del agente puedan ser manipulados de forma inadvertida. Dentro de este conjunto de técnicas anti-manipulación se incluyen otras como son las de cifrado, checksum, anti-depurado, anti-emulación, etc.[31], [32], que en definitiva comparten el mismo objetivo, pero que también están orientadas a la prevención del análisis de la función que implementa el propio agente. Todas estas técnicas presentan un grado de dificultad bastante alto en su aplicación, especialmente para los programadores que utilicen esta tecnología, principalmente debido al hecho de que la aplicación de estas técnicas implica ciertos conocimientos mínimos de seguridad.

Una vez expuestas las diferentes alternativas, defendemos el hecho de que todas las técnicas descritas proporcionan protección a corto plazo, además de requerir una base de conocimiento mínima en temas de seguridad, por lo tanto, no son útiles para nuestro propósito. En cambio, en ciertos escenarios, pueden representar una solución bastante factible, especialmente combinada con otras soluciones. Existen algunas soluciones teóricas al problema, de hecho se ha demostrado que la autoprotección de código no es viable, es decir, que no se puede asegurar la fiabilidad de una solución que sólo se base en software para la protección [36]. En algunos escenarios, la protección requerida está limitada a algunas partes del software (código o datos). De esta forma, la función implementada por el software o los datos han de estar ocultos para el host en el que se ejecute el software. Algunas de estas técnicas requieren un paso de procesamiento adicional externo para obtener los resultados deseados. Entre estos esquemas, las técnicas de ocultación permiten la evaluación de funciones cifradas [37]. Esta técnica se basa en proteger los datos procesados para la protección del propio agente. Por esta razón, es una técnica apropiada para la protección de los agentes. Sin embargo, sólo es aplicable para la protección de funciones polinomiales.

El caso de los esquemas de colaboración en línea también es muy interesante. En estos esquemas, parte de la funcionalidad del software se ejecuta en uno o más computadores externos. La seguridad de esta solución depende de la imposibilidad de cada parte de identificar la función realizada por otros. Esta solución es muy apropiada para arquitecturas de computación distribuidas como son los sistemas basados en agentes o la computación grid.

Por último, existen técnicas que crean una protección bidireccional, protegen tanto al agente como a la agencia. Entre ellas algunas se basan en el uso de soluciones basadas en hardware, como algunas que usan el Trusted Computing Platform como elemento base [39]. De hecho, con la reciente aparición de la computación ubicua, la necesidad de una plataforma segura se ha consolidado como un hecho evidente. Esta solución añade un componente confiable a la plataforma de computación, que normalmente consiste en cierto hardware integrado en la placa madre del ordenador y que se utiliza para crear confianza en los procesos software [38]. Otras técnicas se basan únicamente en elementos software, como por ejemplo la que hace uso de la computación protegida [40]. Esta técnica se basa en la partición de los elementos software en dos o más partes mutuamente dependientes, de forma que el código será ejecutado de forma remota por diferentes agentes.

VI. Conclusiones

La AmI implican un salto desde las aplicaciones que proporcionan cierta funcionalidad y pueden ser utilizadas por una entidad externa, hacia aplicaciones que tienen su propia autonomía y saben exactamente como comportarse en entornos

humanos sin su interacción explícita. Desde un punto de vista abstracto, hemos descubierto las similitudes existentes entre las necesidades de AmI y las primitivas del paradigma del agente. En AmI, cada aplicación de ambiente se comportará como un agente que tiene personalidad y capacidad de decidir. Cada aplicación necesita ser autónoma, reaccionar a los cambios de entorno, y tomar la iniciativa para satisfacer las necesidades humanas de la forma correcta y en el momento adecuado. Las aplicaciones también necesitan incorporarse a un entorno abierto de otras aplicaciones y comunicarse con ellas, por lo que necesitan de algún tipo de habilidad social. Además, consideramos la AmI, que está enfocada a servir de forma impredecible en los escenarios de la vida diaria, como un sistema complejo abierto que necesita de ciertas labores de ingeniería que usan técnicas más avanzadas que las de los sistemas concretos y bien definidos. Defendemos la idea de que el paradigma del agente software es muy prometedor para el desarrollo de los sistemas de AmI durante todas las fases del ciclo de vida del desarrollo y no sólo en la implementación del mismo. Las metodologías de la ingeniería del software orientada a agentes tienen un gran potencial con respecto a la AmI, así que el siguiente paso podría ser adaptar metodologías existentes, o incluso crear alguna nueva, para hacer de la AmI una ingeniería en todas sus fases de desarrollo desde la recolección de requisitos hasta la implementación final.

REFERENCIAS

[1] N. Streitz and P. Nixon. The Disappearing Computer. Communications of The ACM, March 2005/Vil. 48, No.3.

[2] Krogstie, J., et al., Research Areas and Challenges for Mobile Information Systems. International Journal of Mobile Communication, 2004. 2(3).

[3] Krogstie, J. Requirements Engineering for Mobile Information Systems. In Proceedings of the Seventh International Workshop on Requirements Engineering: Foundations for Software Quality (REFSQ'01). 2001. Interlaken, Switzerland.

[4] Eu Project Report. ISTAG Scenarios for Ambient Intelligence 2010. ftp://ftp.cordis.lu/pub/ist/docs/istagscenarios2010.pdf

[5] M. Wooldridge and N. R. Jeanings, Intelligent agents: Theory and practice,íí Knowl. Eng. Rev., vol. 10, no. 2, 1995.

[6] M. Wooldridge. Agent-based Software Engineering. In IEEE Proceedings on Software Engineering, 144(1), pages 26–37, February 1997.

[7] N. Jennings, M, Wooldridge. Applications of intelligent agents. En: Agent technology: foundations, applications, and markets. 1 ed. New York: Springer-Verlag New York, Inc.; 1998:3–28.

[8] Maña A et al. A framework for secure execution of software. International Journal of Information Security. 2004;3(2):99-112.

[9] Paolo Giorgini and B. Henderson-Sellers (Eds.) Agent-Oriented Methodologies, Idea Group Inc., 2005

[10] Philips Research. Ambient Intelligence Research in ExperienceLab. http://www.research.philips.com/technologies/syst_softw/ami/

[11] MIT Ambient Intelligence Research Group. http://ambient.media.mit.edu/

[12] Mark Weiser. The Computer for the Twenty-First Century. Scientific American, pp. 94-10, September 1991.

[13] Remagnino, P. and Foresti, G.L. Ambient Intelligence: A New Multidisciplinary Paradigm. IEEE Transactions on Systems, Man and Cybernetics, Part A, Volume 35, Issue 1, Jan. 2005 Page(s):1 - 6.

[14] A. Schmidt. Implicit Human Computer Interaction Through Context. Personal Technologies Volume 4(2&3), June 2000. pp191-199.

[15] A. Schmidt et al. Advanced Interaction in Context. The International Symposium on Handheld and Ubiquitous Computing (HUC99), Karlsruhe, Germany, 1999 & Lecture notes in computer science; Vol 1707, ISBN 3-540-66550-1; Springer, 1999, pp 89-101.

[16] Anind K. Dey and Gregory D. Abowd. Towards a Better Understanding of Context and Context-Awareness. In the Proceedings of the CHI 2000 Workshop on The What, Who, Where, When, and How of Context-Awareness, The Hague, Netherlands, April 1-6, 2000.

[17] Jolle Coutaz et al. Context is key, Communications of the ACM, v.48 n.3, March 2005

[18] K. Lyytinen , Y. Yoo The Next Wave of Nomadic Computing: A Research Agenda for Information Systems Research. Sprouts: Working Papers on Information Environments, Systems and Organizations. Vol. 1, Issue 1, Article 1 - 2001.

[19] N.R. Jennings. On Agent-Oriented Software Engineering. Artificial Intelligence 117 (2) 277-296 (2000).

[20] P. Bresciani et al. TROPOS: An Agent-Oriented Software Development Methodology. Journal of Autonomous Agents and Multi-Agent Systems. Kluwer Academic Publishers Volume 8, Issue 3, Pages 203 - 236, May 2004.

[21] J. Mylopoulos, L. Chung, and E. Yu. From Object-Oriented to Goal-Oriented Requirements Analysis. Communications of the ACM, 42(1):31ñ37, Jan. 1999.

[22] J. Mylopoulos, Information modeling in the time of the revolution, Information Systems, v.23 n.3-4, p.127-155, May 1, 1998

[23] E. Yu. Agent Orientation as a Modelling Paradigm. Wirtschaftsinformatik. 43(2) April 2001. pp. 123-132.

[24] A. Perini et al. A Knowledge Level Software Engineering Methodology for Agent Oriented Programming. In the Proceedings of the Fifth International Conference on Autonomous Agents, Montreal, Canada - May 29 - June 01, 2001.

[25] G.Booch .Object-Oriented analysis and design with applications.Addison Wesley (1994)

[26] M. Wooldridge, N. R. Jennings, and D. Kinny. The Gaia Methodology for Agent-Oriented Analysis and Design. In Journal of Autonomous Agents and Multi-Agent Systems. 3(3):285-312. 2000.

[27] A. Dardenne, A. van Lamsweerde and S. Fickas. Goal-Directed Requirements Acquisition. Science of Computer Programming Vol. 20, North Holland, 1993, pp. 3-50.

[28] S. Berkovits, J. Guttman, V. Swarup. Authentication for Mobile Agents. Mobile Agents and Security, Springer-Verlag publishers vol.1419, 1998, pp114-136.

[29] G.Necula. Proof-Carrying Code. Proceedings of 24th Annual Symposium on Principles Programming Languages, 1997.

[30] Yee, Bennet S. A Sanctuary for Mobile Agents. Secure Internet Programming, 1999.

[31] I. Schaumaller-Bichl, E.Piller. A Method of Software Protection based on the used of Smartcards and Cryptographic techniques. Proceedings of the Eurocrypt, Springer-Verlag LNCS 0209, pp 446-454, 1984.

[32] J.P. Stern et al. Robust Object Watermarking: Application to Code. Proceedings of Info Hiding, Springer-Verlag, LNCS 1768, pp 368-378, 1999.

[33] C. Collberg, C. Thomborson. Watermarking, Tamper-Proofing, and Obfuscation-Tools for Software protection. University of Auckland technical report 170, 2000.

[34] P. Wayner. Dissapearing Cryptographiy. Information Hiding, Stenography and Watermarking. Morgan Kauffman 2002.

[35] B. Barak et al. On the (Im)possibility of Obfuscating Programs. Proceedings of CRYPTO. Springer-Verlag, LNCS 2139, pp1-18, 2001.

[36] O. Goldreich. Towards a theory of software protection. Proceedings of the 19th Ann. ACM Symposium on Theory of Computing, pp 182-194, 1987.

[37] T. Sander, C.F. Tschudin. On Software Protection via Function Hiding. Proceedings of Information Hiding. Springer-Verlag, LNCS 1525, pp 111-123, 1998.

[38] S. Pearson et al. Trusted Computing Platforms. Precntice Hall 2003.

[39] A. Muñoz, A. Maña, D. Serrano. The Role of Trusted Computing in the Secure Agent Migration. In International Journal of Computer Science & Applications. ISSN 0972-9038.

[40] A. Maña, A. Muñoz, D. Serrano. Towards Secure Agent Computing for Ubiquitous Computing and Ambient Intelligence. Fourth International Conference, Ubiquitous Intelligence and Computing, Hong Kong (China) 2007. LNCS.

EVADIR: una metodología para la evasión de IDS de red

Sergio Pastrana
Escuela Politécnica Superior
Universidad Carlos III de Madrid
Email: spastran@inf.uc3m.es

Agustín Orfila
Escuela Politécnica Superior
Universidad Carlos III de Madrid
Email: adiaz@inf.uc3m.es

Arturo Ribagorda
Escuela Politécnica Superior
Universidad Carlos III de Madrid
Email: arturo@inf.uc3m.es

***Abstract*—Los sistemas de detección de intrusiones de red o *Network Intrusion Detection Systems* (NIDS) son herramientas software o hardware que monitorizan el tráfico de red en busca de actividad maliciosa de distinta naturaleza. A medida que aparecen nuevas vulnerabilidades y métodos para explotarlas, estos sistemas son convenientemente actualizados. En caso de producirse, los ataques son detectados si un NIDS actualizado está monitorizando el sistema atacado, exceptuando aquellos ataques cuya firma aún no ha sido desarrollada (ataques de día cero). Esta situación ha provocado que los atacantes traten de desarrollar nuevas técnicas para pasar desapercibidos a ojos del NIDS cuando intentan atacar un sistema, o lo que es lo mismo, tratan de evadir su detección. Al igual que ocurre con las nuevas vulnerabilidades, los NIDS deben estar preparados para contrarrestar estas novedosas técnicas evasivas. Actualmente la mayoría de estas técnicas se basan en explotar ambigüedades presentes en los protocolos de red, problema que fue presentado por Ptacek y Newsham [7]. En este artículo presentamos EVADIR, una nueva metodología que facilita la búsqueda de nuevas formas de evadir NIDS. El núcleo principal de la misma consiste en modelar el NIDS mediante Programación Genética (PG), haciendo uso de una sintaxis más sencilla que la del NIDS. De este modo, se facilita la búsqueda de técnicas evasivas sobre el detector al realizar el análisis sobre el modelo generado, cuya semántica es menos compleja que la del NIDS en cuestión.**

I. Introducción

Las tecnologías de la información se han convertido en un componente crítico de la economía actual. La protección de las mismas frente a acciones hostiles determinará, en gran medida, el desarrollo de la sociedad de la información y las comunicaciones. Las medidas de seguridad se suelen clasificar atendiendo a su forma de actuación en: de prevención, detección, corrección y recuperación. Aunque las más aconsejables de las anteriores son las primeras, el hecho de ser más costosas que las restantes, junto con la imposibilidad de anular completamente un riesgo, hace preferible distribuir los recursos entre todas las medidas citadas, dando lugar así a la llamada defensa perimetral, consistente en la construcción de sucesivas barreras de protección: preventivas, detectoras, correctivas y recuperadoras.

Los sistemas de detección de intrusiones son herramientas software o hardware que automatizan el proceso de monitorizar los eventos que acontecen en un ordenador o red en busca de evidencias de intrusiones [20]. Una intrusión se define como un intento de comprometer la confidencialidad, la integridad, o la disponibilidad de la información, o de soslayar los mecanismos de seguridad de un ordenador o red. Los IDS de red o NIDS toman como fuente de información principal el tráfico de red. A este respecto, se define el concepto de evasión sobre un NIDS como aquella técnica que permite transformar un ataque definido y detectable en otra forma del mismo que lo soslaye. Es decir, el paquete o conjunto de paquetes (en caso que haya fragmentación o que el ataque se encapsule en una sesión entera) que contengan el ataque es modificado con el fin de que el NIDS no sea capaz de procesarlo adecuadamente, y por lo tanto, de detectarlo.

Los NIDS son, junto con los cortafuegos, los primeros mecanismos de seguridad usados para proteger a los sistemas de información en red. En consecuencia son objetivos prioritarios de los atacantes, que tratan de deshabilitarlos o forzarlos a producir información errónea. En general, los NIDS no proporcionan una respuesta automatizada a las intrusiones que detectan, y se precisa del análisis por parte de un administrador de seguridad de las alertas emitidas para determinar el alcance de las mismas y establecer qué medidas correctoras o recuperadoras han de llevarse a cabo. Por este motivo, la información proporcionada por los NIDS, en caso de ser falsa, puede dar lugar a la distracción del personal de seguridad, como si de un señuelo se tratara [10].

Las técnicas de evasión propuestas hasta la fecha se basan, fundamentalmente, en el abuso de las ambigüedades existentes en los protocolos de comunicaciones. Dichas ambigüedades pueden provocar que los NIDS y los sistemas finales a proteger interpreten los protocolos de distinta forma, provocando que el procesamiento de las conexiones por parte de los NIDS y de los equipos finales sean diferentes. Esta situación favorece el diseño de ataques que evadan a los NIDS.

La investigación en técnicas de evasión de NIDS basados en usos indebidos es, en conjunción con el descubrimiento y detección de nuevas formas de ataque, el principal modo de mejorar la eficacia de estos. En la actualidad, la rápida actualización de las firmas de los NIDS para la detección de nuevos ataques de red provoca que los atacantes traten de desarrollar técnicas evasivas, más sigilosas y difíciles de detectar. Así, un administrador de seguridad no es consciente de que el NIDS ha sido evadido hasta el posterior análisis forense de los sistemas comprometidos. Este contexto es la principal motivación de este trabajo, cuyo objetivo primordial es establecer una metodología que permita el descubrimiento

de nuevas formas de evadir NIDS y que, en consecuencia, favorezca el desarrollo de mecanismos en los NIDS para su prevención.

Para la consecución de este objetivo, en la metodologia propuesta EVADIR (EVAsión de la Detección de Intrusiones en Red) se generan modelos (mediante PG) que tratan de emular el comportamiento de NIDS existentes con el fin de interpretar su modo de funcionamiento desde una perspectiva alternativa. Estos modelos, además de simplificar la complejidad de los NIDS, permiten modelar el comportamiento de NIDS de código cerrado, de los que se desconoce su modo de operación interno. Su análisis facilita el hallazgo de nuevas formas de evasión, difíciles de encontrar por otros medios. La aplicación práctica de esta metodología puede favorecer el diseño de NIDS más difíciles de evadir.

El resto del artículo se estructura de la siguiente manera. En la sección II se presenta cual es el estado actual de la línea de investigación en las técnicas evasivas y el uso de PG en el ámbito de los NIDS. En la sección III se explica formalmente en qué consiste la metodología propuesta, así como las tareas que la componen. Finalmente, en la sección IV se establecen las conclusiones.

II. Estado de la cuestión

La evasión de NIDS fue tratada por primera vez por Ptacek y Newsham en 1998 [7]. En este artículo, los autores resaltan dos grandes problemas en el procesamiento del tráfico de red por parte de los NIDS. En primer lugar, un NIDS no siempre es capaz de conocer con exactitud la manera en que los paquetes serán procesados en el sistema final, debido a las ambigüedades existentes en los protocolos TCP/IP. Por ejemplo, algunas implementaciones descartan los segmentos TCP cuyo campo *checksum* sea erróneo, pero otras no. Si el NIDS descarta un paquete con dicho campo erróneo, se expone a que el paquete sea procesado en el sistema final, y viceversa, puede procesar el paquete cuando en el sistema final no se procesaría. Un distinto procesamiento de los segmentos TCP puede conllevar que ciertos segmentos de un ataque no sean procesados (y por lo tanto detectados) por el NIDS, o puede darse que se procesen más paquetes de la cuenta, llegándose a un reensamblado distinto en los NIDS y en los sistemas monitorizados. El segundo problema presentado es que es posible realizar ataques de Denegación de Servicio (DoS) sobre los NIDS. Un envío de muchos paquetes mal construidos hacia un NIDS puede provocar que éste genere numerosas alertas, llegando incluso a no procesar algunos paquetes debido a la sobrecarga. Una atacante podría, antes de enviar los paquetes que componen un ataque, provocar esta situación, por lo que ésta sería la antesala del verdadero ataque que pasará inadvertido.

Con el fin de explotar y estudiar las ambigüedades mencionadas anteriormente, se han diseñado algunas herramientas específicas, como Fragroute, la cual intercepta, modifica, y reescribe el tráfico destinado a un sistema [4], o IDSprobe, que genera tráfico modificado a partir de trazas originales [8]. Las modificaciones que estos sistemas proponen están orientadas a la generación de tráfico que provoque evasiones. De esta manera, se pueden realizar auditorías sobre NIDS haciendo uso de técnicas evasivas existentes. Alternativamente a estas herramientas, se pueden utilizar algunas funcionalidades de programas más conocidos como Nikto [13] o Nmap [12], cuyos parámetros de ejecución permiten modificar los paquetes que se envían para intentar evitar la detección por parte de los NIDS.

La literatura existente sobre técnicas que prevengan a los NIDS ante posibles evasiones se centra en el tratamiento del tráfico que le llega al NIDS, con el fin de paliar las ambigüedades en la interpretación de los protocolos de comunicaciones y establecer un procesamiento común. Watson et al. [1] proponen un sistema para depurar paquetes de datos, de manera que se eliminen los posibles intentos de evasión de los NIDS, e implementan un depurador (*scrubber*) sobre el protocolo TCP para generar tráfico bien formado a partir de uno de entrada que pueda contener ambigüedades. Por su parte, Handley et al., proponen el concepto de normalizadores de tráfico [9]. Éstos son elementos que se sitúan como intermediarios en la red y que toman el tráfico antes de que llegue al NIDS con el fin de eliminar las posibles ambigüedades que puedan ocurrir. Dado que algunos de los problemas que pueden llevar a evasiones se dan en el reensamblado de los paquetes, estos normalizadores deben almacenar el estado y los paquetes previos de todas las conexiones entrantes, lo que requiere una gran capacidad de almacenamiento, ya que deben verificar la consistencia de los paquetes subsecuentes. Esta tarea consume una gran cantidad de recursos (tanto de procesamiento como de almacenamiento) y puede convertirse en un cuello de botella cuando se trabaja con redes de alta velocidad [2].

También se han propuesto algunas soluciones alternativas que no precisan de la modificación del tráfico. Varghese et. al. [3] plantean dividir la firma que busca el NIDS para detectar el ataque (NIDS *signature*) en pequeños fragmentos, de manera que cualquier paquete que contenga cualquiera de los fragmentos sería detectado de forma rápida, pasando después el conjunto de paquetes a un análisis más lento pero más profundo y por lo tanto más eficaz. Shankar y Paxson [5] proponen un sistema que notifica al NIDS la topología de red y el comportamiento (entendiéndose como tal la implementación de los protocolos) del sistema final que se monitoriza, de manera que pueda modificarse la configuración del NIDS para ajustarse a la información proporcionada por su sistema. A este respecto Snort [6], uno de los NIDS de código abierto más utilizado en la actualidad, implementa una técnica similar a esta en el preprocesador IP (frag3) de su última versión. Finalmente, Antichi et al. [11] proponen el uso de Bloom Filters, una suerte de filtros distribuidos que consiguen que la firma del NIDS sea detectada sin necesidad de un reensamblado previo de los paquetes. Con este sistema se minimizan los falsos negativos pero se incrementa sobremanera la carga computacional del NIDS.

En lo referente a la búsqueda de evasiones en NIDS, cabe destacar el trabajo de Mutz et al. [27], en el cual

los autores proponen realizar ingeniería inversa (tomando un conjunto de entradas/salidas y analizando el comportamiento de los ficheros binarios de un NIDS comercial) con el fin de obtener información acerca de las firmas que aplican los NIDS cuyo comportamiento no es conocido al ser el código propietario. Una vez analizado su funcionamiento, los autores presentan un ejemplo de evasión con un ataque a servidores Apache. Su trabajo se centra en la evasión de las firmas, las cuales son el último escalón en la arquitectura de un NIDS, sin tener en consideración el preprocesamiento que pueda realizarse anteriormente. Sin embargo, las técnicas evasivas propuestas por Ptacek y Newsham [7] se basan precisamente en ambigüedades de este preprocesamiento.

La idea principal de nuestra propuesta, alternativa e innovadora frente a los antecedentes citados, consiste en desarrollar una metodología que facilite la búsqueda de técnicas de evasión de NIDS. Para la consecución de este objetivo se construirán modelos, que emulen el comportamiento de NIDS existentes. Para ello se utilizará programación genética [24], la cual se ha mostrado como un paradigma eficaz y eficiente de cara al desarrollo de NIDS [14], [15], [16], [17], [18], [19], lo que avala su utilización para el objetivo en cuestión.

III. EVADIR: Metodología para la evasión de NIDS

Este artículo tiene como objetivo principal la propuesta de una nueva metodología para buscar formas de evadir NIDS que permita predecir y prevenir futuros intentos de evasión. Se plantean las siguientes grandes tareas a cumplimentar dentro de la metodología (ver Figura 1):

1) Generar un conjunto de datos que represente adecuadamente la idiosincrasia del tráfico de red actual. Dicho tráfico incluirá tráfico normal y malicioso (incluyendo técnicas evasivas existentes).
2) Definir y diseñar un sistema basado en Programación Genética que aborde el modelado automático de NIDS en producción y que proporcione una representación simplificada pero fiel de los mismos.
3) Desarrollar nuevas técnicas evasivas analizando el modelo generado.

A. Generación del tráfico y de los conjuntos de entrenamiento y test

1) Preparación de la infraestructura virtual: En primer lugar se define la arquitectura lógica de red que se utilizará en el entorno de pruebas. Ésta ha de ser fiel a un entorno real para que el tráfico que se genere sea similar al que podríamos encontrar en los sistemas de información y las organizaciones. Dicha arquitectura (lógica) se compone una serie de equipos para la auditoría, para la instalación de los NIDS y para la máquina que hace las funciones de víctima. La arquitectura dispone además de un canal de comunicación que conecta las máquinas junto con uno o varios routers. Una vez definida la arquitectura lógica, se establece una arquitectura física haciendo uso de la técnica de virtualización [25] ya que ofrece una mayor flexibilidad y no requiere de una gran infraestructura física. El requisito aquí es disponer de equipos

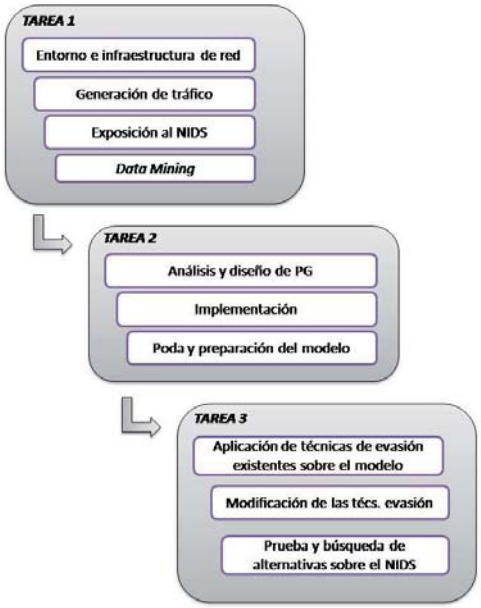


Fig. 1. Arquitectura de EVADIR.

de gran potencia y almacenamiento con varias tarjetas de red que permitan realizar la simulación de la infraestructura. Como software de virtualización, se utiliza alguno de los existentes de código abierto, como VMWare o VirtualBox, ya que permite reducir costes y cumple con los requisitos necesarios en esta tarea.

En esta fase, se instalan y configuran las máquinas virtuales usadas para la adquisición de datos. Han de instalarse tanto los Sistemas Operativos (varios, con el fin de trabajar con NIDS que funcionen sobre distintas plataformas) como el software necesario. La elección del S.O. y el software auxiliar a utilizar es una cuestión que debe definirse en el momento de implementar la metodología.

2) Generación y etiquetado del tráfico: Haciendo uso del entorno virtual configurado, se genera tráfico variado, entendiéndose como tal el tráfico normal (es decir, simples conexiones TCP, UDP e IP sin actividad maliciosa), de ataque (los paquetes llevan encapsulados algún tipo de ataque, sobre todo de tipo HTTP) y evasivo (modificación los paquetes de ataque y normal haciendo uso de las técnicas de evasión existentes en la literatura, las cuales están implementadas y documentadas, [7]). La generación del tráfico se realiza de forma controlada, es decir, estableciendo de manera concisa qué paquetes se envían y filtrando la captura por la máquina auditora, de manera que se garantiza que el tráfico cumple con las características deseadas. Además, los datos se acompañan con una serie de metadatos para etiquetar y documentar las trazas generadas, ya que como se verá más adelante, este tráfico se utiliza en varias etapas de esta metodología.

Para generar el tráfico, se hace uso de herramientas de

código abierto existentes que permiten el envío de segmentos TCP con la suficiente flexibilidad para modificar las cabeceras de los mismos y de los paquetes IP. De esta manera se pueden desarrollar las técnicas evasivas descritas por Ptacek y Newsham y generar así el tráfico evasivo. El contenido del tráfico enviado es algún ataque detectado en principio por cualquier NIDS (como por ejemplo, un ataque XSS sobre un servidor web) cuando se quiera generar tráfico de ataque, o simples peticiones HTTP no maliciosas cuando se quiera generar tráfico normal. Se ha decidido descartar la utilización de conjuntos de datos etiquetados públicamente disponibles en la literatura como LBNL [21] o KDD [22] debido a que el primero sólo incluye tráfico de reconocimiento (y además anonimizado) y el segundo data del año 1999 y su validez despierta controversia [23].

La generación del tráfico es una tarea crucial para la obtención de resultados útiles dentro de esta metodología, ya que va a determinar todos los procesos subsequentes como se verá más adelante. Debe por lo tanto garantizarse, mediante el análisis del tráfico generado, que todos los casos que se pretenden abordar están realmente reflejados en el tráfico, de lo contrario, ha de repetirse esta tarea hasta que los objetivos queden totalmente satisfechos.

3) Exposición del tráfico a los NIDS: El tráfico generado se presenta de manera *offline* a los NIDS bajo estudio con el fin de obtener y registrar la salida que éstos dan. Es posible que parte del tráfico evasivo no sea detectado, por lo que se deben analizar las posibles evasiones que se produzcan. Es de esperar en esta fase que el tráfico normal pase desapercibido en las alertas de los NIDS, y que el tráfico malicioso sea detectado, lleve o no incluídas técnicas de evasión, ya que éstas están documentadas y existen herramientas que las tratan (e.g. Frag3 de Snort).

4) Procesamiento y extracción de características del tráfico: Hasta este punto lo que se ha obtenido es un conjunto de paquetes de tráfico sin tratar junto con la salida que los NIDS estudiados ofrecen cuando dicho tráfico se les presenta de manera *offline*. En este punto, se debe generar un conjunto de datos que relacione los paquetes de tráfico obtenidos con su correspondiente salida dada por cada NIDS. Dado que muchos NIDS trabajan a nivel de conexión y no de paquete, será necesario para esos casos constituir las conexiones presentes en el tráfico junto con la salida dada por el NIDS para esas conexiones.

Debido al gran volumen de datos generado, para modelar NIDS se precisa de una fase de preprocesamiento de los datos, haciendo uso de técnicas de Minería de Datos (*Data Mining*) con el fin de tomar las características del tráfico más relevantes desde la perspectiva de la seguridad, constituyéndose posteriormente sendos conjuntos de entrenamiento y de test. El conjunto de datos está constituido por entradas como por ejemplo las que se muestran en la Figura 2. En esta figura, se pueden ver una serie de campos separados por comas, con 2 etiquetas (los 2 últimos campos). Estas son, un símbolo para denotar si el paquete se corresponde con una actividad benigna (B en el ejemplo) o maliciosa (M en el ejemplo), y otro símbolo que denota si el NIDS ha acertado o no en la predicción (en el ejemplo, - indica que lo etiqueta bien, y + que lo etiqueta mal).

```
22,37273,54637,56676,5,0,0,0,1,1,0,58464,28642,0,M,+
37273,22,56676,54637,5,0,0,0,0,1,0,34800,52234,0,M,-
22,2224,54113,20593,5,0,0,0,1,1,0,32760,44444,0,B,-
1068,139,59831,24387,5,0,0,0,1,1,0,15376,64779,0,B,+
139,1068,24387,59831,5,0,0,0,1,1,0,17520,62547,0,B,-
22,37273,54637,56676,5,0,0,0,1,1,0,58464,28482,0,M,+
37273,22,56676,54637,5,0,0,0,0,1,0,34800,52154,0,M,-
22,37273,54637,56676,5,0,0,0,1,1,0,58464,28402,0,M,+
22,37273,54637,56676,5,0,0,0,1,1,0,58464,28322,0,M,+
22,2224,54113,20593,5,0,0,0,1,1,0,32760,44716,0,B,-
22,2224,54113,20593,5,0,0,0,1,1,0,32760,44580,0,B,-
1417,1103,61855,25411,5,0,0,0,0,1,0,17010,53159,0,B,-
2224,22,20593,54113,5,0,0,0,1,1,0,17460,59804,0,B,+
1103,1417,25411,61855,5,0,0,0,1,1,0,64450,5683,0,B,-
2224,22,20593,54113,5,0,0,0,1,1,0,17400,59728,0,B,-
```

Fig. 2. Ejemplo de trazas que se obtendrán como conjuntos de entrenamiento y test

B. Modelado de los NIDS

1) Análisis y diseño de la arquitectura de la Programación Genética: Primero se determinan cuales son los elementos terminales, operadores, función de fitness, parámetros de ejecución del algoritmo, profundidad máxima del árbol, etc. Los operadores y terminales determinan la semántica de los modelos generados, por lo que se eligen de forma que faciliten la consecución del objetivo principal establecido, esto es, con el fin de obtener un modelo fácil de interpretar que permita la búsqueda de evasiones sobre él. Para la obtención de los valores óptimos de los parámetros del algoritmo se utiliza la técnica de validación cruzada [26]. El funcionamiento es el siguiente:

1) Dividir el conjunto de entrenamiento en k hojas.
2) Establecer aleatoriamente unos valores para los parámetros que se deseen configurar. Se recomienda acotar dichos valores para facilitar el proceso de búsqueda (por ejemplo, no se debería permitir que el valor del número de generaciones fuera menor de 10).
3) Para cada k
   - Generar un fichero que contenga todas las trazas menos la de la hoja k.
   - Entrenar con ese fichero.
   - Testear con el fichero que contenga las trazas de la hoja k.
4) Almacenar las medias de los resultados de test.
5) Volver al paso 2. Este proceso se repetirá al menos $4n$ veces, donde n es el número de parámetros a configurar, para asegurarse una buena variabilidad.

Una vez finalizada la ejecución, se obtiene la configuración de parámetros que mejor resultados haya dado para utilizarla en la búqueda del modelo. La función de fitness en un principio será el error de clasificación, ya que no tiene sentido hablar de tasas de falsos positivos o negativos al tratarse de un modelado

del comportamiento de un NIDS, no de una detección real de ataques. Sin embargo, esta función podrá ser cualquier otra que se estime oportuno.

2) Implementación de la búsqueda del mejor modelo: En esta fase se busca el individuo (mediante PG con los parámetros determinados en B.1) que mejor emula a cada NIDS sobre el conjunto de entrenamiento. Los resultados obtenidos sobre el conjunto de test determinan si es necesario redefinir el diseño para obtener un mejor individuo. Es posible por lo tanto que esta tarea y la anterior estén correlacionadas, es decir, en el momento de analizar los resultados, si se observa que no son los esperados o deseados, es necesario que redefinir la arquitectura de PG, estableciendo nuevos parámetros, generando nuevos operadores o redefinendo la función de fitness.

3) Poda y preparación del modelo: En esta fase, se realiza una optimización manual del modelo para ajustarlo a una semántica de interpretación sencilla por parte de un humano. Esto es, el árbol generado mediante PG puede no ser sencillo de interpretar, puede usar nomenclaturas complejas, o quizás tenga nodos o incluso ramas enteras redundantes.

C. Análisis y estudio de técnicas de evasión

1) Aplicación de técnicas de evasión existentes: Se aplican las técnicas de evasión existentes en los paquetes de ataque para presentárselo al modelo generado y así poder observar su salida. Esto permite saber si el modelo es capaz o no de detectar los ataques cuando éstos han sido modificados para intentar evadir a los NIDS. El comportamiento observado se compara con el que había dado el NIDS original, para ver si realmente este modelo es fiel a su comportamiento. En esta fase se podrán extraer conclusiones acerca de qué técnicas evasivas no funcionan sobre el NIDS original y sí sobre el modelo generado, lo cual no es de mucha ayuda al ser este modelo en principio inexacto (es decir, no se comporta exactamente igual al NIDS). Sin embargo, si se diera alguna evasión en el NIDS original pero no sobre el modelo generado, podrían extraerse conclusión acerca del porqué el NIDS falla, y por lo tanto se podría ofrecer una solución.

2) Modificación de las técnicas de evasión existentes: Una vez estudiado el comportamiento del modelo ante las técnicas evasivas, se buscan otras nuevas para evadirlo. Esto se hace modificando las técnicas evasivas originales con el fin de buscar una manera distinta pero relacionada que, aplicadas sobre el modelo generado, provoquen que éste no sea capaz de detectar el ataque.

Es en esta tarea donde se corrobora la utilidad de modelar los NIDS, ya que los modelos generados nos permiten, mediante una sintaxis más sencilla y definida previamente, analizar el comportamiento que éstos tienen sin tener que entrar en los detalles internos de su implementación. Estos detalles, como se ha comentado previamente, pueden ser demasiado complejos y en muchos casos inaccesibles (cuando el código no sea abierto), por lo que la búsqueda de técnicas evasivas se convertiría en una tarea similar a la fuerza bruta. Sin embargo, utilizando los modelos obtenidos, esta tarea se realiza de una forma más sencilla y eficaz.

3) Prueba de las técnicas modificadas sobre el NIDS original y búsqueda de alternativas: Aquellas técnicas de evasión modificadas que realmente evadan el modelo generado son aplicadas al NIDS original con el fin de estudiar si las modificaciones han provocado también la evasión en éste. Es de esperar que, si el comportamiento del modelo es suficientemente parecido al del NIDS, aquellas evasiones que sean exitosas sobre el modelo lo sean también sobre el NIDS.

Hasta este punto de la metodología las evasiones probadas se basan en técnicas existentes o modificaciones de ellas. En este punto se van a probar nuevas técnicas, de distinta naturaleza, que, basándose en la semántica menos compleja del modelo evadan tanto al modelo como al NIDS (lo cual es el objetivo prioritario). Esta parte de la metodología es la menos automatizada, y la más estocástica, ya que, en función de la calidad del modelo, y aplicando la filosofía de prueba y error, encontrar nuevas evasiones puede no ser una tarea sencilla. Para poder encontrar nuevas técnicas, es necesario un conocimiento exhaustivo tanto del comportamiento del modelo (el cual lo ofrece el propio modelo) como del comportamiento de los protocolos de red (el cual está documentado y estudiado), por lo que es en esta parte en la cual el investigador ha de poner una mayor atención, ya que el resto de la metodología es un proceso automatizado que, una vez definido, es fácilmente replicable.

En la Figura 3 se puede observar un diagrama general del flujo de información en EVADIR. Como se puede observar, el proceso de búsqueda de evasiones se realiza primero sobre el modelo generado, y una vez que se encuentran técnicas para evadirlo, éstas se aplican sobre el NIDS original para evaluar su comportamiento.

IV. Conclusión

En los últimos años la complejidad de las Tecnologías de la Información se ha visto severamente incrementada. Esta complejidad hace que los sistemas requieran cada vez más de técnicas que garanticen su seguridad. Los Sistemas de Detección de Intrusiones de Red (NIDS) basados en usos indebidos son herramientas que detectan posibles intrusiones en un sistema, basándose en ataques conocidos y para los que están preparados. Cada vez que se publica un nuevo ataque, los administradores actualizan las firmas de sus NIDS para que sean capaces de detectar ese ataque. Esta situación provoca que los atacantes busquen nuevas técnicas para evitar ser detectados cuando pretenden comprometer un sistema. Los NIDS basados en la detección de usos indebidos detectan cualquier ataque para el que estén configurados siempre y cuando sean capaces de procesar los paquetes que encapsulan dicho ataque. Sin embargo, existe la posibilidad de evadir este procesamiento [7]. En este artículo se ha presentado una metodología que automatiza el proceso de buscar nuevas técnicas evasivas mediante el modelado del comportamiento de los NIDS a través de técnicas de Programación Genética. Los modelos generados tienen una semántica más sencilla

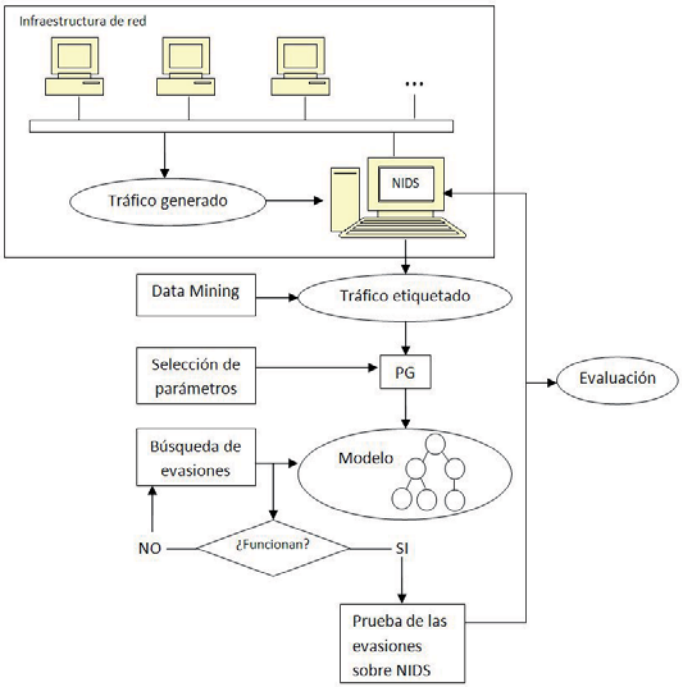


Fig. 3. Diagrama de flujo de información en EVADIR

que los NIDS, por lo que sobre estos modelos es más fácil buscar técnicas evasivas. A este respecto, cabe destacar que los NIDS propietarios no publican su código y por lo tanto no es sencillo conocer su funcionamiento interno [27], por lo que un modelado del mismo es adecuado para el estudio de su comportamiento. Es de suponer que, siempre y cuando el proceso de modelado se haya realizado de una manera correcta, las técnicas evasivas que funcionen sobre el modelo también lo hacen sobre el NIDS original.

Hasta la fecha hemos realizado pruebas de concepto sobre un NIDS basado en un árbol de decisión, haciendo uso de un escenario en el que hay presente tanto tráfico normal como tráfico malicioso (concretamente, escaneos de puertos). Los resultados de la prueba son fructíferos, ya que se han logrado encontrar algunas técnicas evasivas novedosas. Actualmente estamos aplicando esta metodología sobre NIDS reales, tanto de código abierto como propietarios. Como trabajo futuro se automatizará el proceso de buscar evasiones sobre los modelos (actualmente las evasiones son buscadas analizando la semántica del modelo producido manualmente), para lo cual se hace necesario establecer un formato común de la sintaxis que utilizan éstos modelos.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente realizado en el marco del proyecto SEGUR@, subvencionado por CDTI, Ministerio de Industria, Turismo y Comercio de España, dentro del programa CENIT, con referencia CENIT-2007 2004

REFERENCES

[1] D. Watson, M. Smart, R. G. Malan, and F. Jahanian, "Protocol scrubbing: network security through transparent flow modification", en *IEEE/ACM Transactions on Networking*, vol. 12, pp. 261–273, 2004.

[2] M. Vutukuru, H. Balakrishnan, and V. Paxson, "Efficient and Robust TCP Stream Normalization",en *SP '08: Proceedings of the 2008 IEEE Symposium on Security and Privacy*, Washington, DC, USA, 2008, pp. 96–110.

[3] G. Varghese, J. A. Fingerhut, and F. Bonomi, "Detecting evasion attacks at high speeds without reassembly",en *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, Pisa, Italy, 2006, pp. 327–338.

[4] D. Son. (2002) Fragroute. [Online]. *HYPERLINK "http://www.monkey.org/ dugsong/fragroute/"*

[5] U. Shankar and V. Paxson, "Active Mapping: Resisting NIDS Evasion without Altering Traffic",en *SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*, Washington, DC, USA, 2003, p. 44.

[6] M. Roesch, "Snort - Lightweight Intrusion Detection for Networks", en *LISA '99: Proceedings of the 13th USENIX conference on System administration*, Seattle, Washington, 1999, pp. 229–238.

[7] T. H. Ptacek and T. N. Newsham, "Insertion, evasion and denial of service: Eluding network intrusion detection", Technical report, 1998.

[8] L. Juan, C. Kreibich, C.-H. Lin, and V. Paxson, "A Tool for Offline and Live Testing of Evasion Resilience in Network Intrusion Detection Systems",en *DIMVA '08: Proceedings of the 5th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Paris, France, 2008, pp. 267-278.

[9] M. Handley, C. Kreibich, and V. Paxson, "Network intrusion detection: Evasion, traffic normalization and end-to-end protocol semantics", en *Proceedings of the 10th Conference on USENIX Security Symposium*, Volume 10, 2001, p. 9.

[10] D. J. Chaboya, R. A. Raines, R. O. Baldwin, and B. E. Mullins, "Network instrusion detection", vol. 8, pp. 36-42, 2006.

[11] G. Antichi, D. Ficara, S. Giordano, G. Procissi, and F. Vitucci, "Counting Bloom Filters for Pattern Matching and Anti-Evasion at the Wire Speed", en *IEEE Network Magazine of Global Internetworking*, vol. 23, no. 1, pp. 30-35, Feb. 2009.

[12] Nmap. [Online]. *HYPERLINK "http://nmap.org/"*

[13] Nikto. [Online]. *HYPERLINK " http://www.cirt.net/code/nikto.shtml"*

[14] A. Orfila, J. M. Estevez-Tapiador, and A. Ribagorda, "Evolving High-Speed, Easy-to-Understand Network Intrusion Detection Rules with Genetic Programming", en *EvoWorkshops '09: Proceedings of the EvoWorkshops 2009 on Applications of Evolutionary Computing*, Tübingen, Germany, 2009, pp. 93–98.

[15] J. Blasco, A. Orfila, and A. Ribagorda, "Improving Network Intrusion Detection by Means of Domain-Aware Genetic Programming",en *Proceedings of the 5th International Conference on Availability, Reliability and Security ARES 2010*, Krakow, Poland, 2010.

[16] G. Folino, C. Pizzuti, and G. Spezzano, "GP Ensemble for Distributed Intrusion Detection Systems",en *ICAPR, 2005*, pp. 54-62.

[17] S. Mukkamala, A. Sung, and A. Abrham, "Modeling intrusion detection systems using linear genetic programming approach", en *IEA/AIE'2004: Proceedings of the 17th international conference on Innovations in applied artificial intelligence*, Ottawa, 2004, pp. 633–642.

[18] S. Peddabachigari, A. Ajith, C. Grosan, and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems", en *Journal in Network Computer Applications*, vol. 30, no. 1, pp. 114-132, 2007.

[19] M. Crosbie and E. Spafford, "Applying Genetic Programming to Intrusion Detection", en *Working Notes for the AAAI Symposium on Genetic Programming*, 1995, pp. 1-8.

[20] R. Bace and P. Mell, "NIST Special Publication on Intrusion Detection Systems," 800-31, 2001.

[21] Lawrence Berkley National Laboratory and ICSI. (2005) LBNL/ICSI Enterprise Tracibng Project. [Online]. www.icir.org/entreprise-tracing/

[22] S. Hettich and S. Bay. (1999) The UCI KDD Archive. [Online]. http://kdd.ics.uci.edu

[23] N. Athanasiades, J. G. Levine, H. L. Owen, and G. F. Riley, "Intrusion Detection Testing and Benchmarking Methodologies",en *Proceedings of the International Information Assurance Workshop*, IWIA 03, Maryland, 2003, pp. 63-72.

[24] J. R. Koza, "Genetic Programming: On the Programming of Computers", M. Press, Ed.Cambridge, MA, USA, 1992.

[25] P. Barham , B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, ”Xen and the art of virtualization”, en *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, October 19 - 22, 2003, ACM, New York, NY, 164-177.

[26] R. Kohavi, ”A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”, in *IJCAI: International Joint Conference on Artificial Intelligence*, Volume 2, Issue 1, 1137–1143, 1995

[27] D. Mutz, C. Kruegel, W. Robertson, G. Vigna, and R. A. Kemmerer ”Reverse Engineering of Network Signatures”, in *Proceedings of the AusCERT Asia Pacific Information Technology Security Conference, Gold*, 2005

High-speed free-space quantum key distribution system for urban applications

M. J. Garcia*, D. Soto, N. Denisenko, D. Arroyo, AB. Orue and V. Fernandez
Instituto de Física Aplicada
Consejo Superior de Investigaciones Cientificas
*Corresponding author: mariajose.garcia@iec.csic.es

***Abstract*—A high-clock-rate free-space quantum key distribution (QKD) system at a wavelength of 850 nm is presented. The system is designed to securely transmit cryptographic keys at high transmission rates between two locations in urban areas. The pointing, acquisition and active tracking of the QKD system are also presented.**

Keywords-**Quantum key distribution, quantum cryptography, optical data communications, optical fibre, free space communications, data encryption.**

I. Introduction

The need for alternative approaches to ensure worldwide data protection has become a priority during the last two decades, since increasingly more sensitive information is exchanged and more users are involved. In this sense Quantum Information represents a new paradigm in the Information Age, where quantum properties such as the Heisenberg Uncertainty Principle, the No-Cloning Theorem or Entanglement have become the cornerstone in information security. Likewise the advances in quantum computing [1][2] consolidate the idea of future powerful quantum computers capable of superfast parallel computation, jeopardising classical methods or protecting cryptographic keys based on public key cryptosystems. In this scenario quantum cryptography [3] is taking an increasingly important role, as it allows secure transmission of keys even against a quantum computer attack. Moreover, a quantum computer attack would have a retrospective effect, making data protected in the past also vulnerable. In this sense quantum cryptography should not be considered only as a technology for the future, but also a technology to ensure the secrecy of information encrypted *today* that we wish to maintain secret in the future. Although quantum keys have been transmitted at considerable distances such as 200 km [4] and 144 km [5] using fibre optic and free space links respectively, the key exchange rates of these links are still prohibitive low. Therefore the crucial challenge remains the distribution of keys at high speeds for Quantum Key Distribution (QKD) to represent a real and efficient alternative to classical key distribution systems.

Another challenge arises from the impossibility of using quantum repeaters, since a fully functional practical *quantum repeater* [6] is still beyond current technology and the delicate quantum states encoding the key would be perturbed. Therefore to habilitate key exchange among users located arbitrary on the globe, satellite-based links have been studied. In this sense research on free-space QKD systems has been especially focused on increasing the transmission distance of the free-space links in locations situated far from urban areas, designed to emulate satellite-to-earth links [7]. However, much less attention has been paid to short-distance high-transmission rate QKD systems for urban areas.

Nowadays metropolitan fiber-optic networks suffer from the so-called 'connectivity bottleneck', referred to an imbalance located in many parts of the network caused by requirements of flexibility and cost effectiveness of service provisioning. Possibly the most viable alternative for addressing this bandwidth shortage is Free-Space Optics (FSO) [8]. Compared to fiber optic, FSO provides more flexibility and ease of deployment in multiple architectures, and therefore an economic advantage over optical fibre. Applied to QKD, FSO offers the possibility to establish not only earth-satellite but also high-rate short links. As mentioned above, the first option would allow global implementation of QKD whereas the second would enable high bit rate secure communication in metropolitan networks with a high-bandwidth demand. This would be attractive for commercial and financial buildings that wish to be connected to the backbone network. Therefore the development of short-range high bit rate QKD systems in urban areas is an increasing demand.

We are building a GHz-clocked free-space QKD system that implements the B92 protocol [9]. This protocol uses only two non-orthogonal states of a quantum system as opposed to four, like in the BB84 protocol, since two states are enough to implement secure QKD. In practice, these two states can be two linearly polarised states at a nonorthogonal angle, as it is implemented in our system. Alice then, encodes the binary levels '1' and '0' in the two polarised states, and sends them to Bob. When the receiver performs projections onto subspaces orthogonal to the signal states, he can measure the bits with certainty at the expense of some loss. This loss is the effect of the Heisenberg Uncertainty Principle, as nonorthogonal states cannot be distinguished unambiguously without perturbation. After the transmission Bob tells Alice in which instances he detected a photon. In this case there is no need for reconciliation of basis sets between Alice and Bob to discriminate unambiguous measurements, as opposed to the BB84 protocol, which makes this protocol simpler and faster to execute. However, the B92 protocol is particularly vulner-

able to the "intercept-resend" eavesdropping attack. Eve can substitute the transmission channel for a perfectly transparent one and resend the photons to Bob, i.e., she uses this 'lossless' channel to hide the loss she introduces in the measurement. This eavesdropping attack is especially harmful, as Eve would not introduce an additional error to the Quantum Bit Error Rate (QBER)- a measure of how 'secure' the transmission has been. However, the problem of simulating a noisy channel between Alice and Bob while extracting information of it is far from trivial [10]. Moreover, recently the security of the B92 protocol has been proven even in the presence of a lossy channel [11].

II. Realisation of the QKD free-space system

The proposed short-distance QKD system is shown in Figures 1 and 3. The main goal is to securely transmit quantum keys between two locations in Madrid situated at a distance of 3 km at high clock frequencies. These two locations are the Institute of Applied Physics, of the Spanish National Research Council (CSIC) and the Spanish telecommunications operator Telefonica.

A. Realisation of the QKD transmitter: Alice

Figure 1 shows the layout of the transmitter optics. To achieve high transmission rates Alice will use a fast GHz pre-programmed pulse pattern generator, able to generate pseudo-random sequences up to $2^{31}-1$ bits in length, in conjunction with $\lambda \sim 850$ nm vertical-cavity surface emitting lasers (VCSELS) controlled by high-speed drivers. The output of each laser is sent to a collimator by a single-mode at $\lambda \sim 850$ nm optical fibre. Two high extinction-ratio polarisers are used to generate the polarisation states required by the B92 protocol. These states are combined by a non-polarising beam splitter cube and the beam is then focused by a short focal length achromatic lens. Two high-quality mirrors lead the beam to a second achromatic lens which expands the beam to a diameter of $\sim$ 40 mm. After the second lens the beam is ready to be transmitted to the receiver, Bob. The collimators, combining cube, mirrors and lenses are mounted on a 30 cm-side square platform. This platform is mounted on a high-precision gimbal system, which will be used for the alignment of emitter and receiver.

The transmitter mounted on the gimbal system is shown in Figure 2. It consists of a structure which provides with the movements of Right Ascension (RA) and DEClination (DEC) with high precision. Once the initial or *coarse pointing* is performed, which will be discussed in section III, only two variables in spherical coordinates will be enough for the alignment of both stations. The motors employed have precision of microradians, have large loads capacities and are controlled by a programmable double-axis driver capable of implementing alignment and/or tracking algorithms.

The motor in the base provides the RA movement while the lateral motor the DEC movement. In each lateral support a bearing holds a rod with an L-shaped holder where Alice is mounted on.

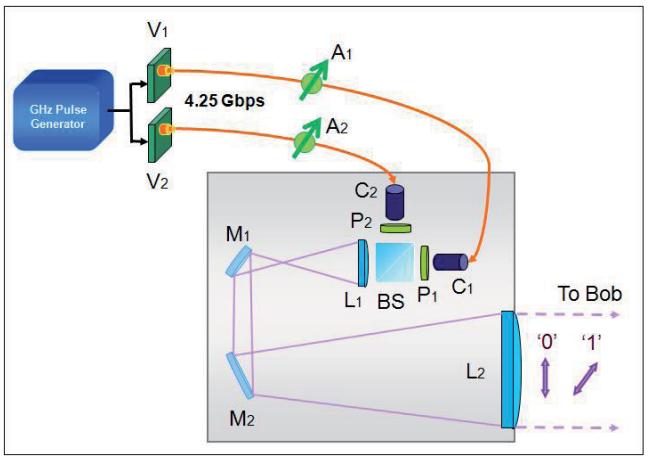


Fig. 1. Diagram of the transmitter: Alice. V_1 and V_2 are two VCSELs; A_1 and A_2 are two fibre-optic attenuators; C_1 and C_2 are two fibre-coupled collimators; P_1 and P_2 are two high extinction-ratio polarisers; BS is a non-polarising beamsplitter; L_1 and L_2 are two achromatic doublet lenses; M_1 and M_2 are two high-reflectivity mirrors.

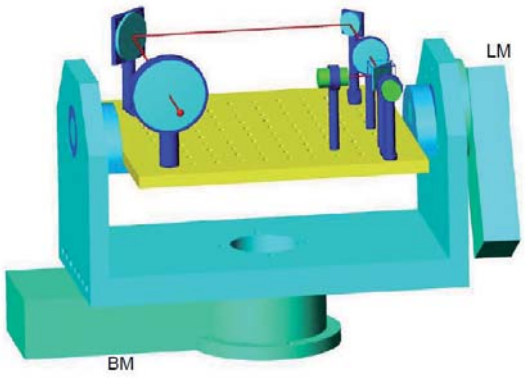


Fig. 2. AutoCAD design of the gimbal system for the transmitter. Two high precision DC motors - a lateral and a base motor represented in the figure by LM and BM - will provide the tip/tilt movements required for the alignment and tracking of Alice and Bob.

B. Realisation of the QKD receiver: Bob

Figure 3 shows the layout of the receiver. A Schmidt-Cassegrain telescope at Bob will efficiently focus the beam coming from Alice. After the telescope, a CCD camera will ease the alignment of emitter and receiver. Bobs optics has been designed to be coupled to the output of the telescope by using SM1 (25 mm) mounts, since they are light-weight and compact. Especial care must be paid to one of the most critical parts of the system, which is the filtering of the solar background radiation from the sun. For this purpose, a combination of spectral and spatial filters will be used. The spatial filtering will be carried out by optical fibre. A good compromise of the diameter of this fibre must be found, as small diameters improve the filtering of the solar radiation at the expense of higher signal losses. In addition, if the diameter is too small the signal could be lost due to the beam wandering caused by the fluctuations of the index of refraction of the air.

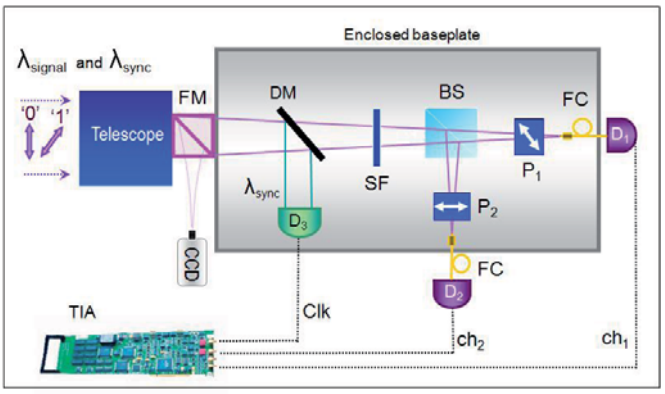


Fig. 3. Diagram of the receiver: Bob. Two different wavelengths will be used: λ_{signal} to transmit the quantum key, and λ_{sync} to time stamp Alice and Bob. FM is a Flip Mirror; DM is a Dichroic Mirror; SF is a Spectral Filter; BS is a 50-50 non-polarising beamsplitter; P_1 and P_2 are two high extinction-ratio polarisers; FC is a fibre patch cord; D_1 and D_2 are single photon detectors; D_3 is an avalanche photodiode; and TIA is a Time Interval Analyser card.

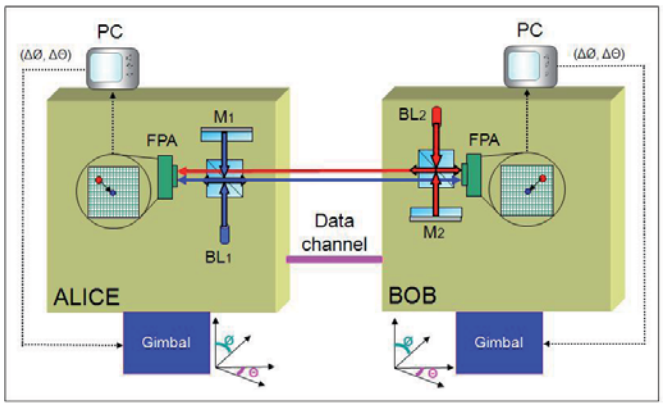


Fig. 4. Tracking of the QKD system. Two computers (PC) will control both the transmitter and receiver's gimbals providing tip/tilt movements, $\Delta\theta$ and $\Delta\phi$, required for the alignment and tracking of both stations. Two Focal Plane Arrays (FPA) in the transmitter and receiver will detect any deviation of the position of both beacon lasers, BL_1 and BL_2, and order the motors to compensate for them. M_1 and M_2 are two mirrors.

As previously mentioned, the states '1' and '0' are encrypted by the two non-orthogonal polarising states used in the QKD protocol B92. These two states will be discriminated by using a non-polarising beamsplitter and two high-extinction polarisers. Two silicon Single-Photon Avalanche Diodes (SPADs) will be used to detect the photons from Alice. In addition, a high speed Time Interval Analyser (TIA) will detect the arrival time of each of the photons that reach the receiver and a software will process the received optical signal and establish the degree of security in the transmission. The electronic card acquired for this purpose measures the photons arrival time with high temporal precision and is able to perform 3.5 millions of measurements per second with a timing jitter lower than 70 ps. The arrival times that this card provides will serve to reconstruct the signal sent by Alice and determine whether the transmission of the key has been secure or not, i.e. the QBER. The synchronisation timing of Alice and Bob will be performed by a different wavelength than that used for key transmission (λ_{sync} and λ_{signal} respectively in Figure 3). This will be discussed in detail in section IV.

III. Pointing, Acquisition and Tracking

Ideally a FSO link should be capable of reliable pointing, acquisition and tracking to offer optimal performance. The first, also known as *coarse pointing*, involves acquiring the approximate location of the receiver by using some prior information and aiming the transmitter in the proper direction. *Spatial acquisition* refers to the operation the receiver does to determine the direction of arrival of the transmitter's beam. Once coarse pointing and spatial acquisition are performed, emitter and receiver have determined their Line of Sight (LOS). *Spatial tracking* refers to the operation of maintaining this LOS with minimum error throughout the transmission. Coarse pointing can be performed by several means. If the distances involved are only a few kilometres, GPS or optical imaging systems can be used. The first uses the stamps from several satellites to compute the location of the target. However, this suffers from lack of satellite accessibility at certain times or locations. Optical imaging consists of visualising the target using lenses or CCD cameras. In this case, the advantage comes from the full accessibility of the cameras at any time.

Spatial tracking requires from both the emitter and receiver to make adjustments to maintain the alignment of the optical beam. One method involves the use of beacon lasers in combination with Focal Plane Arrays (FPA) [12][13] (see Figure 4). A Focal Plane Array is an image sensing device made of a rectangular array of light-sensitive pixels that reproduce a real image according to the intensity of light received by each pixel. When Alice and Bob are aligned an image of the beams is recorded by the FPA and the distance between the spots is computed. At this position both beams are parallel. Therefore any change in the relative position of the spots means the beams are no longer parallel and the computer-driven high-precision gimbal motors receive a signal to counteract this misalignment. For this method to work effectively Alice needs to send her data signal parallel to her beacon laser. Both beams can be generated by different lasers or else Alice can use her data laser as a beacon laser but this means that part of her data signal will be sacrificed for the alignment of the system.

If no tracking is performed large beam diameters must be used for the transmission at long distances in order to minimise the effects of atmospheric turbulence and mechanical vibrations of the equipment on the beam alignment. However larger beam diameters need big telescopes to efficiently detect them and usually a considerable part of the signal is lost. In QKD systems, optical amplifiers or quantum repeaters are still beyond current technology and therefore losses must be avoided by all means. Hence tracking becomes a necessity in this kind of systems.

In summary, along with a properly collimated beam, a high precision mounting platform for Alice is required in order to effectively perform the pointing of the system. An actively

controlled mechanism will be used to compensate for any misalignment of the optical beam to maintain the line of sight. In our system, Alice and Bob will be both mounted on gimbals and the coarse pointing will be performed by two CCD cameras. Fine pointing and active tracking will be performed by aligning both gimbals with the help of two beacon lasers, one from Alice to Bob and the other from Bob to Alice. The wavelength of the beacon lasers will be different from that used by the data laser, making both channels independent. Both wavelengths will be separated by the help of dichroic and interference filters at both ends. Once the beacon beams are aligned, the tracking mechanism will detect any deviation and consequently send a signal to each gimbal driver to realign the beams.

IV. Synchronisation and QBER analysis

The method we have chosen to synchronise Alice and Bob utilises a periodic bright pulse of a different wavelength from that used for the key, as precursor to open a time gate for the subsequent signal photons. This timing pulse will be sent at a different wavelength in parallel with the quantum channel at 850 nm. Both wavelengths will be separated in Bob by a dichroic mirror and interference filters. The frequency of the synchronisation or timing pulse will be a sub-multiple of the clock frequency. At the receiver, the arrival time of the timing pulse is utilised as an arm signal that activates a timing gate in which QKD data is expected. We believe this type of synchronisation in conjunction with a GHz-clocked source will result in secure key transmission rates considerably higher than those currently achieved.

Moreover, a software algorithm that determines the QBER and corrects for additional errors is being programmed. As stated by Shannon's theorem [14] a minimum of bits must be sent via the classical channel to correct the errors. Recently, Low Density Parity-Check Codes (LDPC) codes have been proposed along with a simple iterative decoding algorithm [15]. These codes have been demonstrated to perform very close to the Shannon limit. The advantage of LDPC compared to standard Cascade error correction algorithms, is that LDPC has the potential of a faster implementation as it requires only one round.

V. Conclusions

There is a growing demand for more bandwidth in certain regions of metropolitan networks due to lack or poor connections. High-speed links using free-space optics is an attractive solution and QKD free-space links can offer both speed and security to this problem.

In this line, a high-bit-rate free-space QKD system for urban-span secure communication links has been designed. To achieve high speed transmission a high-clock-frequency pulse generator combined with high-data transmission laser diodes and drivers at Alice has been utilised. Moreover, an optical synchronisation at a different wavelength will permit faster key generation than those currently achieved. In addition, the proposed tracking system controlled by high-precision motors will allow continuous operation of the presented QKD system.

Acknowledgement

We would like to thank the Ministerio de Educación y Ciencia, proyecto MTM2008-02194 and CDTI, Ministerio de Industria, Turismo y Comercio (Spain), in collaboration with Telefónica I+D, Project SEGUR@ with reference CENIT-2007 2004.

References

[1] L. DiCarlo, J. M. Chow, J. M. Gambetta, Lev S. Bishop, B. R. Johnson, D. I. Schuster, J. Majer, A. Blais, L. Frunzio, S. M. Girvin and R. J. Schoelkopf, "Demonstration of two-qubit algorithms with a superconducting quantum processor", *N*ature 460, 240 (2009).

[2] D. Hanneke, J. P. Home, J. D. Jost, J. M. Amini, D. Leibfried and D. J.Wineland, "Realization of a programmable two-qubit quantum processor", *N*ature Physics 6, 13 (2010).

[3] C. H. Bennett and G. Brassard, "Quantum cryptography: public key distribution and coin tossing", *P*roc. IEEE Int. Conf. Comput., Syst. Signal Process., Bangalore, India, 175 (1984).

[4] H. Takesue, S. W. Nam, Q. Zhang, R.H. Hadfield, T. Honjo, K. Tamaki and Y. Yamamoto, "Quantum key distribution over a 40-dB channel loss using superconducting single-photon detectors", *N*ature Photonics 1, 343 (2007).

[5] T. Schmitt-Manderbach et al., "Experimental demonstration of free-space decoy-state quantum key distribution over 144 km", *P*hysical Review Letters 98, 010504 (2007).

[6] H.-J. Briegel, W. Dür, J. I. Cirac and P. Zoller, "Quantum repeaters: The role of imperfect local operations in quantum communication", *P*hys. Rev. Lett. 81, 5932 (1998).

[7] C. Kurtsiefer, P. Zarda, M. Halder, H. Weinfurter, P. M. Gorman, P. R. Tapster and J. G. Rarity, "Quantum cryptography: a step towards global key distribution", *N*ature 419, 450 (2002).

[8] H. Willebrand H and B. S. Ghuman,"Free Space Optics: Enabling Optical Connectivity in Today's Networks", ISBN 978-0-672-32248-8, SAMS, USA, (2002).

[9] C. H. Bennet, "Quantum cryptography using any two nonorthogonal states", *P*hysical Review Letters 68, 3121 (1992).

[10] K. Tamaki, M. Koashi and N. Imoto, "Security of the Bennett 1992 quantum key distribution protocol against individual attack over a realistic channel", *P*hysical Review A 67, 032310 (2003).

[11] K. Tamaki, N. Lütkenhaus, M. Koashi and J. Batuwantudawe, "Unconditional security of the Bennet 1992 quantum-key-distribution scheme with a strong reference pulse", *P*hysical Review 80, 032302 (2009).

[12] A. Portillo, G. G. Ortiz and C. Racho,"Fine pointing control for optical communications", *I*EEE Aerospace Conference Proceedings 3, 1541 (2001).

[13] M. Jeganathan, A. Portillo, C. Racho, S. Lee, D. Erickson, J. De Pew, S Monacos and A. Biswas, "Lessons learnt from the Optical Communications Demonstrator (OCD)", *P*roc. SPIE 3615, 23 (1999).

[14] C. E. Shannon, "A mathematical theory of communication", *B*ell System Technical Journal 27, 379 (1948).

[15] A. Shokrollahi, "LDPC codes: an introduction", *C*oding, Cryptography and Combinatorics 23, 85 (2004).

Acceso seguro a redes de sensores en SCADA a través de Internet

Cristina Alcaraz, Rodrigo Roman, Pablo Najera, Javier Lopez
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga
Email: alcaraz, roman, najera, jlm@lcc.uma.es

***Resumen*—Las Infraestructuras Críticas (ICs) son monitorizadas por sistemas altamente complejos, conocidos como sistemas SCADA (Sistemas de Control y Adquisición de Datos), cuyo principal soporte se encuentra en las subestaciones, las cuales miden de primera instancia el estado real de tales ICs. Para mejorar este control, la industria está actualmente demandando la integración en el modelo tradicional de dos avances tecnológicos: Internet y las redes de sensores inalámbricas. Sin embargo, su incorporación requiere analizar los requisitos de seguridad que surgen en dicho contexto, así como diversos aspectos correlacionados (ej. mantenimiento, rendimiento, seguridad y optimización) y, en base a estos, la estrategia de integración más adecuada para satisfacer dichos requisitos. Este artículo proporciona dicho análisis en profundidad con el fin de ofrecer un modelo de integración seguro adecuado para entornos críticos.**

***Index Terms*—Sistemas Críticos de Control, Sistemas SCADA, Redes Mesh Inalámbrica de Sensores, el Internet, Internet of Things.**

I. Introducción

La introducción de nuevas tecnologías y diferentes tipos de sistemas de comunicación en las redes de control industriales está impulsando nuevos e importantes avances en los procesos de automatización y control. Un caso particular son los SCADA que emplean nuevas tecnologías para monitorizar en tiempo real muchas de las infraestructuras críticas (ICs) desplegadas en nuestra sociedad, tales como los sistemas de energía, de transporte o distribución de agua/aceite. Específicamente, en estos momentos dos de las tecnologías más demandadas son las redes inalámbricas e Internet. El primero, dado que proporciona los mismos servicios de control que una infraestructura cableada, pero con un bajo coste de instalación y mantenimiento. El segundo, al ofrecer conectividad global independientemente de la posición física de los dispositivos, tales como nodos sensores configurados en las subestaciones para controlar las infraestructuras críticas.

La imagen 1 muestra un sistema SCADA actual [1], donde los operadores autenticados y autorizados gestionan los flujos de datos transmitidos por las subestaciones. Una subestación remota se compone de dispositivos de campo, conocidos como Unidades Terminales Remotas (RTUs), capaces de recolectar, gestionar y transmitir los flujos de datos recibidos de sus sensores. Por otra parte, la imagen muestra también nuevas tecnologías adoptadas recientemente por las subestaciones, tales como redes de sensores inalámbricas (Wireless Sensor Networks o WSNs). Este tipo de red es una de las tecnologías

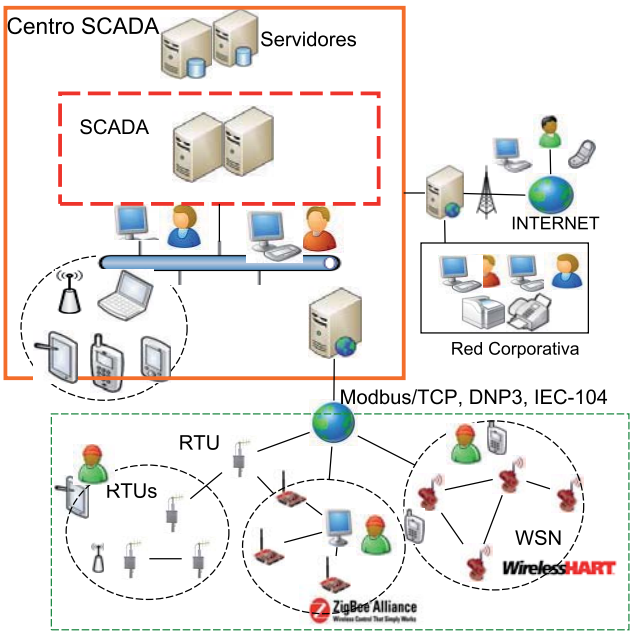


Figura 1. A current SCADA network architecture

más demandadas por los ingenieros industriales, dado que ofrece servicios de control similares a una RTU, pero con un bajo coste de instalación. Sin embargo, tales servicios no están siendo todavía explotados apropiadamente, dado que los estándares de comunicación únicamente contemplan conectividad local. Debido a esto, tanto la industria como la comunidad científica están tratando de maximizar esfuerzos para ofrecer tales servicios a través de Internet. Como resultado, un nuevo paradigma comienza a emerger en el contexto de las infraestructuras críticas, la Internet de los Objetos (IoT).

La IoT está formada de diversas infraestructuras heterogéneas de comunicación interconectadas, donde Internet, los servicios y objetos físicos juegan un importante rol en los procesos de control y automatización. El interés por abrir los procesos de comunicación en ICs a la red de redes y la inminente expansión de los nuevos paradigmas de comunicación ha motivado el desarrollo de diversos trabajos de investigación. Así, Li et. al propusieron en [2] un sistema basado en web para RTUs inteligentes con capacidad para

interpretar HTTP, Jain et. al presentó en [3] un sistema experto basado en web para diagnóstico y control de sistemas de energía e incluso algunas compañías comerciales tales como Yokogawa [4] o WebSCADA [5] están ya proporcionando soluciones de control utilizando Internet.

En particular, las WSNs, como parte de los objetos de la IoT, pueden crear una capa virtual, autónoma e inteligente sobre el entorno físico de subestaciones remotas, proporcionando información sobre el estado del mundo real que puede ser accedido en cualquier momento y lugar. De hecho, los gobiernos de alrededor del mundo han previsto el potencial de las WSNs en infraestructuras críticas y las han incluido en sus planes nacionales para investigación y desarrollo, tal como el gobierno australiano, a través de su Research Network for a Secure Australia (RNSA) [6], o el gobierno de los Estados Unidos en sus planes de protección para ICs [7],[8]. La comunidad científica e industrial está realizando diversas investigaciones para la adopción de las WSN en CIP. Por ejemplo, Bai et al. [9] ha implementado las WSN en un sistema SCADA para la monitorización de la energía generada por una planta de energía eólica. Carlsen et. al. introdujeron en [10] una WSN capaz de predecir la pérdida de aceite/gas en una planta submarina en el Mar del Norte.

La interacción de las WSN en las ICs a través de Internet se puede lograr empleando múltiples estrategias de integración: desde nodos sensores que implementen la pila TCP/IP y se conviertan en miembros completos de la IoT, a redes capilares que mantengan su independencia, pero empleen los servidores de Internet como interfaz hacia las entidades externas. Sin embargo, este camino presenta diferentes problemas que no han sido aún estudiados en profundidad en la literatura, tales como qué estrategia de integración debería emplearse en la integración de las WSN industriales en IoT, qué problemática de seguridad surgirá debido a esta evolución de la arquitectura de red y cómo asegurar que los requisitos de seguridad de los sistemas críticos se satisfacen en este paradigma de red. El objetivo de este artículo es proporcionar una base para la respuesta a estas cuestiones, analizando los requisitos de seguridad e infrastructurales de las WSN industriales conectadas a Internet y discutiendo la adecuación de las estrategias de integración que harán realidad la visión de gestión ubicua en el área de las redes industriales.

El artículo se organiza de la siguiente manera. La sección II describe los requisitos que deben ser considerados para alcanzar una integración segura. La sección III presenta las estrategias de integración susceptibles de ser adoptadas. La sección IV proporciona una análisis de la integración entre WSN e Internet en el contexto de las redes de control dados los requisitos mencionados previamente. La sección V concluye el artículo y muestra las líneas de trabajo futuro.

II. Requisitos de WSN industriales

Con objeto de proporcionar sus servicios, los sensores industriales inalámbricos podría beneficiarse sustancialmente de su integración en la IoT. La colaboración y agregación de datos críticos entre sensores geográficamente dispersos se vería mejorada, proporcionando información más fiable y precisa. Más allá, tanto los operadores de sistemas como los usuarios finales (con privilegios restringidos) podrían beneficiarse del acceso en tiempo real desde cualquier lugar a la infraestructura reduciendo costes. Sin embargo, a pesar de que es posible utilizar diferentes estrategias para conectar las WSNs a Internet, es necesario conocer cuál es más adecuada para los requisitos de cada escenario. El objetivo de esta sección es introducir tanto los requisitos específicos de las WSN industriales antes de presentar las diferentes estrategias de integración.

II-A. Requisitos de Control y Automatización

Para estudiar la seguridad de las WSN industriales en el contexto de Internet, es esencial considerar no sólo los requisitos de seguridad, sino también los requisitos que tales redes de control deben satisfacer, tales como mantenimiento, rendimiento del sistema y fiabilidad de los recursos y servicios. El motivo es simple: algunos de estos requisitos tienen una influencia directa sobre los requisitos de seguridad y viceversa, tales como la sobrecarga en memoria o tiempo de respuesta del nodo debido a los mecanismos de seguridad empleados. Debido a ello, esta subsección introduce los requisitos básicos (incluyendo los de seguridad) que deben considerar tanto sistemas de control como industriales .

II-A1. Mantenimiento: Es necesario realizar el mantenimiento del software y hardware de las subestaciones. Para prevenir la aparición de errores, cada dispositivo debe ser debidamente configurado, y deben realizarse tests periódicos de su estado. Además, los componentes software deben estar actualizados con las revisiones críticas, así como añadirse nuevo hardware a la subestación cuando éste es necesario. Por tanto, las propiedades asociadas al mantenimiento son:

- *Direccionamiento*. Es necesario especificar un tipo de identificación única para cada elemento presente en la subestación de forma que sea posible acceder al flujo de datos que éste produce. Esta propiedad se relaciona con cómo se accede a los diferentes identificadores de los dispositivos y quién se encarga de almacenar dichas identidades.
- *Acceso Interno*. Los servicios ofrecidos por los dispositivos que se encuentran en la subestación deben se accecidos de forma local por los operadores de las subestaciones, ya sea por motivos de testeo o de redundancia. Esta propiedad se relaciona con la complejidad actual de acceder a los dispositivos de la subestación de forma local.
- *Mantenibilidad*. Como con cualquier dispositivo, el software de las RTUs deberá ser actualizado debido a optimizaciones o parches de seguridad entre otros. Esta propiedad se refiere al número de dispositivos que deben cambiar con objeto de actualizar la funcionalidad de la subestación.
- *Extensibilidad*. El número de RTUs que puede encontrarse en una subestación concreta cambia a lo largo del tiempo de vida de la infraestructura. Esta propiedad se

relaciona con los cambios totales que deben realizarse en la subestación para incluir nuevo hardware.

II-A2. Fiabilidad: La funcionalidad proporcionada por la subestación debe ser suficientemente fiable para ofrecer sus servicios con unos niveles de calidad concretos. Los flujos de datos provistos por las RTUs deben estar disponibles en todo momento, y cualquier consulta relativa al contenido actual de dichos flujos de datos debe llegar al sistema central tan rápido como sea posible. Consecuentemente, las propiedad asociadas a la fiabilidad son:

- *Disponibilidad*[1]. Los datos producidos por las RTUs deben estar disponibles en todo momento con objeto de reaccionar a situaciones problemáticas y asegurar la integridad del sistema completo. Como propiedad, se dan dos dimensiones de la misma: la fiabilidad (empleando la redundancia del sistema para evitar los puntos únicos de fallo) y la seguridad (existencia de ataques de denegación de servicio y el empleo de mecanismos de sanado para proporcionar los servicios incluso en el caso de ataques/fallos en el sistema).
- *Rendimiento*. La información debe ser recuperada de las RTUs a velocidad suficiente. Como propiedad, el rendimiento se relaciona con las capacidades hardware de los dispositivos de la subestación, además de la velocidad actual de la infraestructura de la red de la subestación, y el número de saltos entre la RTU y el repositorio de datos.

II-A3. Sobrecarga: Es necesario lograr un balance entre el número de recursos disponibles al dispositivo y su coste global. Los dispositivos no deberían recibir una sobrecarga de trabajo, pero tampoco deberían de dedicarse recursos innecesarios. Más allá, aquellos recursos deberían optimizarase para funcionar en el entorno de la subestación. Consecuentemente, las propiedades asociadas con la sobrecarga son:

- *Recursos del Dispositivo*. Con objeto de implementar los diferentes protocolos que proporcionan la funcionalidad central de las subestaciones, tales como DNP3 o WirelessHART, los dispositivos deben usar parte de sus recursos HW y SW. Esta propiedad referencia la cantidad de recursos que se necesitan dentro de un nodo para implementar dichos protocolos.
- *Optimización*. Hay algunos protocolos específicos que se han optimizado para proporcionar la mejor funcionalidad posible en un entorno particular. Esta propiedad se relaciona con la existencia de protocolos específicos de red (tales como WirelessHART), que son conscientes de las características específicas del entorno de red y utilizarlos para proporcionar mejores servicios (por ej. redundancia de red y robustez del enlace).

II-A4. Seguridad: La seguridad de los diferentes procesos de una subestación es materia de máxima importancia. Cualquier problema que afecte a la integridad de los elementos de una subestación tendrá potencialmente una influencia sobre el mundo real, afectando no sólo a las infraestructuras físicas, sino a los seres humanos. Por lo tanto, sólo usuarios autorizados deben de disponer de privilegios para modificar el estado de los elementos de las subestaciones, y únicamente los usuarios fiables deben poder acceder a los flujos de datos producidos por las subestaciones. De manera adicional, deben existir mecanismos que almacenen las interaciones entre los diferentes elementos, para facilitar no sólo el análisis del comportamiento del sistema, sino también la detección de posibles brechas de seguridad. Por lo tanto, las propiedades asociadas a la seguridad son:

- *Canal Seguro*. Allá donde dos dispositivos que pertenezcan al mismo sistema SCADA (por ej. una máquina del sistema central y una RTU de una subestación)se comuniquen, es importante establecer un canal seguro que soporte servicios de integridad y confidencialidad extremo-a-extremo. La integridad del flujo de datos evitará la introducción de información falsa en el sistema. Además, la confidencialidad del flujo de información evitará el acceso de adversarios a información sensible. Como propiedad, referencia al tipo de máquinas y mecanismos que se ven envueltas en la creación de un canal de comunicación que soporte confidencialidad e integridad.
- *Autenticación*. En lo que se refiere a la autenticación de usuario, los dispositivos deben asegurarse de la identidad de un usuario que solicite una operación concreta. Como propiedad, la autenticación se refiere a la localización y la naturaleza de los mecanismos y elementos que pueden emplearse para proporcionar la identidad de un ser humano.
- *Autorización*. Una vez que cualquier usuario de la red (sea un humano o una máquina) proporciona su identidad, puede ser necesario comprobar si tal usuario tiene los derechos para acceder a la información. No sólo se debe controlar el acceso a la información, sino también la granularidad de la información. Es también necesario monitorizar las operaciones de control (por ej. los dispositivos deben ser sólo reprogramados por los usuarios autorizados). Como propiedad, la autorización referencia a los tipos de mecanismos, credenciales y herramientas que pueden emplearse para comprobar si una cierta entidad está autorizada a realizar una operación.
- *Registro y detección*. Es necesario mantener un registro de las interaciones de los heterogéneos usuarios que acceden a los servicios de una subestación. Tal registro permitirá recrear los incidentes de seguridad y las situaciones anormales. Además, podemos detectar ataques específicos en tiempo real. Como propiedad, el registro y la detección referencian a la estructura de los sistemas de registro y los mecanismos que pueden emplearse para analizarlos.

[1]La disponibilidad puede considerarse como un requisito de seguridad, pero ha sido clasificada como un requisito de fiabilidad debido a su relación cercana a la dimensión funcional de la subestación.

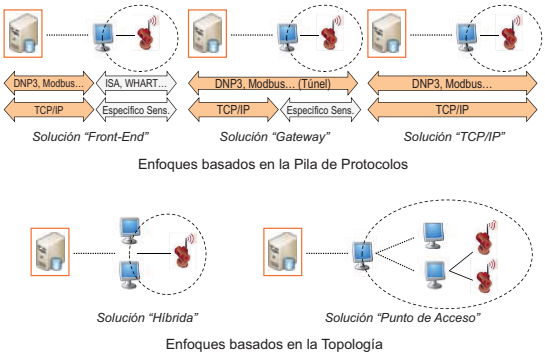


Figura 2. Estrategias de integración

III. Estrategias de Integración

Actualmente existen dos formas de integrar las WSNs industriales en Internet, las cuales van a depender de la (i) *pila de protocolo* o de la (ii) *topología* de la red. Para la primera clasificación, es necesario comprender las similitudes existentes entre ambas tecnologías, obteniéndose tres posibles soluciones/modelos (ver Figura 2): **Front-end** (la WSN es independiente del Internet), **Gateway** (intercambio de información a través de nodos especiales de Internet), y **TCP/IP** (los nodos implementan la pila TCP/IP). Para entrar en más detalle, comentaremos cada uno de estos casos. Por ejemplo, para una solución Front-end, tanto el centro SCADA como las WSNs en las subestaciones remotas no establecen comunicaciones directas, permitiendo a las WSNs tener implementada su propias pilas de protocolos (ZigBee, WirelessHART, ISA100.11.a). En este caso, las interacciones entre ambos extremos deben residir en una interfaz capaz de traducir los respectivos protocolos (ej. una RTU), facilitando la consulta y el control de las subestaciones.

En una solución Gateway, también se considera la existencia de un nodo especial capaz de actuar como un gateway a nivel de aplicación (ej. una RTU), con el fin de traducir los protocolos de las capas inferiores provenientes de ambas redes (ej. TCP/IP y Modbus), además de enrutar la información de un punto a otro (como hace el front-end), sin necesidad de conexión directa con el Internet y sin requerir que la propia pila de WSN cambie. Finalmente, en la solución TCP/IP, los nodos sensores ya forman parte del Internet, al tener implementado dentro de su lógica el estándar TCP/IP o un conjunto de protocolos compatibles (como 6LowPAN [11] en redes 802.15.4 [12]). Esta solución es precisamente la que podría integrar las WSNs industriales en el contexto de IoT (*Internet of Things*).

Por el contrario, en la segunda clasificación, el nivel de integración va a depender de la localización física de aquellos nodos responsables de proveer acceso a Internet, como pueden ser: (i) estaciones bases localizadas en la raíz de un diseño de WSN híbrida (solución **Híbrida**) o (ii) nodos backbone dedicados a proveer puntos de accesos a Internet en un salto (solución **Punto de Acceso**). Las WSNs del primer caso, se caracterizan por ofrecer redundancia cuya información debe pasar a través de ellas. Por el contrario, las redes diseñadas bajo una solución de Punto de Acceso presentan una topología de red en forma de árbol cuyas hojas corresponden a nodos sensores y el resto son considerados puntos de accesos a Internet. Ambas clasificaciones pueden funcionar conjuntamente, permitiendo que nodos backbones o estaciones bases, puedan funcionar como front-ends/gateways, favoreciendo los accesos directos entre los nodos y el centro SCADA. Sin embargo, combinar un modelo de red TCP/IP con soluciones Híbridas/Punto de Acceso puede no tener mucho sentido, al existir en los nodos sensores una vía de acceso directa hacia el Internet.

IV. Análisis de Mecanismos de Integración

Una vez conocidas las diferentes estrategias de integración, es necesario discutir las (des)/ventajas de cada una de ellas en un contexto industrial. Para ello, es necesario considerar las propiedades de la **Sección** II

1) ***Mantenimiento***: En términos de *direccionamiento*, tanto las soluciones Front-end como el Gateway requieren de una tarea de de traducción de identidades a una dirección de un nodo de la red, por lo que el nodo responsable (es decir, la RTU) deberá mantener una tabla de direcciones. En cambio, si la solución es TCP/IP, dicha tabla de direcciones debe ser localizada en el centro SCADA para transmitir directamente hacia el Internet. Luego, existen dos tipos de redes: **descentralizados** o **centralizados**. Obviamente, la gestión en una red centralizada, como una red Híbrida o de Punto de Acceso, será mucho más costosa que la descentralizada al requerirse una replicación de tablas de direcciones en las estaciones bases o backbones o una interfaz intermediaria que permita realizar las traducciones.

Por otro lado, e independientemente de la solución, los operadores tienen que llevar a cabo tareas de mantenimiento mediante conexiones TCP/IP (*acceso interno*) a la propia subestación, con el fin de acceder a los servicios de recuperación de datos. También cabe la posibilidad de que los operadores en campo puedan acceder directamente a servicios locales del propio protocolo de WSN (p. ej. ISA100.11a), tal como ocurre en soluciones Front-end y Gateway. En cambio, en modelos TCP/IP se requiere conocer de antemano la dirección de red del sensor. Por último, esta propiedad puede ser un poco más compleja en soluciones de Punto de Acceso, ya que se necesita que los operadores estén físicamente cercanos a los nodos que deseen acceder.

El *mantenimiento* SW de los nodos sensores va a depender del número de dispositivos a ser actualizados (nuevos servicios SCADA). En soluciones Front-end, el proceso afecta únicamente a un dispositivo de la red (la RTU), el cual requiera parar los procesos de control y la disponibilidad de la WSN momentáneamente con el fin de llevar a cabo las tareas de mantenimiento. En cambio, las soluciones Gateway

y TCP/IP ofrecen actualizaciones graduales en todos los nodos sensores implicados, asegurando continuidad y funcionalidad. Igualmente, la solución Híbrida es también capaz de ofrecer tales actualizaciones graduales al proveer redundancia de elementos, mientras que la solución de Punto de Acceso puede no ofrecerla, ya que los nodos están conectados de alguna forma a un determinado nodo backbone. En lo que respecta a la *extensibilidad*, tanto el Front-end como el Gateway requieren incluir una nueva entrada en la tabla de traducciones para hacer funcionar los servicios específicos de las WSNs, mientras que en soluciones TCP/IP son precisamente los nodos sensores los encargados de añadir nuevas entradas (de manera individual). Igualmente, esta propiedad también podría afectar a tanto soluciones Híbridas como de Punto de Acceso, excepto en el caso particular que dichas tablas sean gestionadas de manera distribuida.

3) ***Sobrecarga***: *Recursos de los dispositivos* están relacionados con todas aquellas soluciones (Híbrida, Punto de Acceso y TCP/IP) que requieran ciertas capacidades SW y HW en los nodos para implementar protocolos de aplicación y servicios de seguridad. Sin embargo, aunque los nodos sensores industriales aparentemente provean ciertas capacidades computacionales y recursos, estos siguen presentando ciertas complejidades (como los nodos convencionales). Con lo que respecta a la *optimización*, todos aquellos modelos (como el Front-end, Gateway, Híbrida y Punto de Acceso) que hacen uso de protocolos específicos de WSN (p. ej. WirelessHART o ISA100.11a), pueden beneficiarse de los servicios ofrecidos por ellos (p. ej. sincronización mediante una determinada TDMA, mecanismos de control de interferencias y coexistencia con otras tecnologías, mecanismos de diagnósticos, gestión de prioridades y gestión de rutas redundantes). La mayoría de estos servicios propios no están contemplados por la pila TCP/IP.

3) ***Seguridad***: Para conseguir un *canal seguro* se requiere mecanismos que aseguren confidencialidad e integridad en todas las comunicaciones, es decir, desde el centro SCADA hasta las WSNs. En el modelo TCP/IP es posible proveer tales servicios, ya que cada nodo final tiene implementado de alguna forma la pila TCP/IP. Incluso, aunque no sea posible hacer uso de las ventajas de IPSec debido a la naturaleza restrictiva de los sensores [13], es posible hacer uso de SSL/TLS implementados en la capa de transporte o WS-ConversacionSegura en la capa de aplicación para aquellas redes que hacen uso de servicios Web. Estos mismos mecanismos de aplicación podrían ser también viables en un modelo Gateway, ya que sus capacidades se centran en reenviar la información. Por el contrario, una solución Front-end necesita proteger el canal de dos formas: (i) con mecanismos de seguridad TCP/IP y (ii) con mecanismos de protección específicos de las WSNs (ej. claves simétricas de WirelessHART). Finalmente, es importante comentar que tanto el modelo Front-end como el Gateway permite implementar una red privada virtual (VPN) entre el centro SCADA y el front-end/gateway, asegurando la confidencialidad y la integridad de los mensajes de control.

Uno de los principales desafíos en lo que respecta a la *autentificación* es determinar la localización de los servicios de autentificación de usuario y el almacenamiento de las credenciales de seguridad, como usuario/contraseña. El modelo más simple es el Front-end, ya que es justo el nodo quién tiene que almacenarlos y gestionarlos. En cambio, estos servicios de seguridad son distribuidos en disversos puntos de la red en las demás soluciones. Una posible solución sería aplicar protocolos y mecanismos de seguridad de manera centralizada con el fin de gestionar en un mismo punto toda la información relativa a la autentificación (ej. Kerberos [14]). No obstante, estas soluciones podrían añadir ciertas complejidades a los nodos sensores al requerir el uso de servicios adicionales para validar las credenciales. Para evitar este hecho, una posible solución sería replicar las bases de datos de credenciales, con el problema añadido de que se podría dificultar los procesos de mantenimiento del sistema completo. Con respecto al modelo Gateway, una medida de seguridad sería configurar canales dedicados y seguros entre el usuario y el gateway justo después de realizarse un proceso de autentificación.

La propiedad de *autorización* es similar a la de autentificación, con la excepción de que es necesario conocer dónde almacenar los servicios asociados a la autorización y los permisos de usuarios. Por consiguiente, las mismas soluciones para la autentificación son efectivas para la autorización. Sin embargo, el uso de bases de datos distribuidas podría incluir una importante complejidad al sistema, ya que los permisos de usuarios tienden a cambiar con mayor frecuencia.

En cuanto al *registro*, es necesario almacenar todas las interacciones entre el sistema central y los nodos sensores. Una medida óptima sería diseñar un modelo de red totalmente centralizado, como podría ser un modelo Front-end y un Gateway, e incluso, si existe mecanismos de seguridad implementados punto a punto, la solución Gateway podría filtrar cierta información manteniendo dicha naturaleza centralizada. En cambio, un modelo descentralizado, como el TCP/IP, podría suponer ciertas complejidades de almacenamiento en los sensores, ya que serían ellos los que deberían registrar todas las interacciones ocurridas en el sistema. Igualmente, la *detección* debería implementarse en soluciones centralizadas, ya que supondría incrementar la inteligencia del nodo con nuevas reglas de detección, reduciendo sus capacidades lógicas para procesar otros tipos de tareas. Este mecanismo se hace prácticamente esencial para ayudar a detectar posibles malfuncionamientos y ataques internos, por lo que se recomienda seguir investigando en la elaboración de reglas y patrones sencillos (no complejos) para WSNs.

IV-A. Discusiones

Considerando los análisis realizados anteriormente, nuestro principal objetivo ahora es determinar la validez de tales estrategias en un contexto industrial. Por ejemplo, en una solución TCP/IP, las WSNs localizadas en subestaciones remotas son consideradas partes del Internet, pero sin embargo, este hecho podría no ser tan ventajoso. En términos de seguridad es necesario proteger la WSN desde cualquier tipo de intruso, ya que un incremento en el tráfico de red podría suponer una

reducción de funcionalidad en los nodos sensores debido a sus limitadas capacidades. La autenticación y la autorización pueden ser resueltas, en un principio, mediante soluciones centralizadas como kerberos. Por otro lado, los nodos que tienen implementado la pila TCP/IP y no tienen suficientes recursos para implementar protocolos específicos WSNs (WirelessHART o ISA100.11a), no llegan a beneficiarse de los servicios de optimización ofrecidos por éstos, e incluso, no pueden implementar mecanismos de almacenamiento y reenvío para ganar redundancia de datos, así como la gestión de datos en caché. No obstante, este modelo podría ofrecer ciertas ventajas en lo que respecta el mantenimiento SW gradual y resistencia a fallos, sin aislar la funcionalidad total del sistema/subsistema.

Por otro lado, los nodos sensores de un modelo Front-end pueden hacer uso de los estándares existentes para implementar mecanismos de seguridad cuya funcionalidad reside en un único punto. Sin embargo, este hecho podría suponer riesgos de aislamiento al tratarse de un punto vulnerable a ataques de denegación de servicios. Una posible solución sería implementar un modelo de red Híbrida o Punto de Acceso, aunque ésto podría suponer nuevos problemas asociados a la replicación de recursos. Otra ventaja de usar Front-end es el uso de servicios óptimos ofrecidos por los estándares específicos de WSNs y la posibilidad de configurar mecanismos de seguridad que ayuden a mejorar la resistencia (p. ej. rutas redundantes). Por último, aunque el mantenimiento SW de la red es simple (actualización en un nodo), existe el riesgo de aislar la funcionalidad de la red durante un periodo de tiempo crucial. Para ello, se recomienda replicar nodos, tal como hace los modelos Híbridos y Punto de Acceso.

En lo que respecta al modelo Gateway, éste puede ser considerado una mezcla entre las dos primeros. En particular, ésta ofrece algunas implementaciones dadas por el Front-end, como puede ser el uso de servicios óptimos específicos de WSN e implementaciones de almacenamiento y reenvío, permitiendo a su vez realizar consultas directas al centro SCADA. No obstante, la funcionalidad del Gateway podría añadir complejidades a los nodos, además de considerar detalles de seguridad como la autentificación y la autorización. Además, el gateway debería filtrar las entradas para analizar las consultas y evitar ataques específicos de aplicación y considerar aspectos como la complejidad añadida en los procesos de mantenimiento SW. Por último, esta solución puede ser combinada con modelos de red Híbridas y de Punto de Acceso para configurar redundancia, aunque es también esencial considerar los problemas relativos a estos modelos, como pueden ser la distribución de tablas y recursos.

Por consiguiente, *parece interesante usar una solución puramente TCP/IP en subestaciones remotas, sin embargo, esta solución podría no ser suficiente para una integración total de WSNs en Internet bajo un contexto industrial*. Además, como los sistemas de control simplemente acceden a flujos de datos y realizan tareas de control bajo comandos, otras soluciones, como el Front-end combinadas con modelos que provean redundancia, podrían ser a priori suficientes para las necesidades actuales de la industria. No obstante, es importante comentar que una integración segura y completa de una WSN en el Internet podría brindar al sistema con nuevos e interesantes mecanismos de control, algunos de los cuales harían uso de servicios Web.

V. Conclusion

Desde que los nodos sensores son parte del IoT e Internet de la Energía, nuevos desafíos de investigación están comenzado a emerger, los cuales están resumidos en este artículo bajo un análisis de integración (segura) de nodos sensores en Internet bajo un contexto industrial. Como conclusión de este análisis vemos que actualmente no es necesario integrar totalmente las redes de sensores industriales dentro de Internet y que una simple red basada en redundancia podría ser, por ahora, suficiente para ofrecer la funcionalidad deseada. Sin embargo, queda pendiente para un futuro investigar cómo explotar el potencial y funcionalidad ofrecido por Internet junto al uso de nodos sensores industriales para así garantizar nuevas e interesantes aplicaciones de control.

Agradecimientos

Este trabajo ha sido parcialmente financiado por ARES (CSD2007-00004), PROTECT-IC (TSI-020302-2009-10) y SPRINT (TIN2009-09237), siendo este último co-financiado por FEDER. El primer y tercer autor han sido financiados por el Ministerio de Educación y Ciencia a través de los Programas F.P.I. y F.P.U. respectivamente.

Referencias

[1] C. Alcaraz, G. Fernandez, R. Roman, A. Balastegui, J. Lopez. *Secure Management of SCADA Networks*. New Trends in Network Management, CEPIS, vol. IX, no. 6, 2008, pp.22–28.

[2] D. Li, Y. Serizawa and M. Kiuchi, *Concept design for a Web-based supervisory control and data-acquisition (SCADA) system*, Transmission and Distribution Conference and Exhibition, Asia Pacific. IEEE/PES , vol. 1 , pp. 32-36, 2002.

[3] M. Jain, A. Jain and M. Srinivas, *A web based expert system shell for fault diagnosis and control of power system equipment*, Condition Monitoring and Diagnosis, pp.1310-1313, ISBN: 978-1-4244-1621-9, 2008.

[4] Yokogawa, http://yokogawa.com/scd/fasttools/scd-scada-websuper-en.htm, accessed on Aplil, 2010.

[5] WebSCADA, http://www.webscada.com/, accessed on April, 2010.

[6] D. Bopping, *CIIP in Australia*, 1st CI2RCO Critical Information Infrastructure Protection conference, Rome, 2006.

[7] The Department of Homeland Security, Science and Technology Directorate, *The National Plan for Research and Development in Support of Critical Infrastructure Protection*, Washington, D.C., 2005.

[8] The Department of Homeland Security, *National Infrastructure Protection Plan Partnering to enhace protection and resilence*, 2009.

[9] X. Bai, X. Meng, Z. Du, M. Gong and Z. Hu, *Design of Wireless Sensor Network in SCADA system for wind power plant*. Automation and Logistics (ICAL), pp. 3023-3027, 2008.

[10] S. Carlsen, A. Skavhaug, S. Petersen and P. Doyle, *Using wireless sensor networks to enable increased oil recovery*, IEEE International Conference on Emerging Technologies and Factory Automation, pp. 1039-1048, 2008.

[11] G. Montenegro, N. Kushalnagar, J. Hui and D. Culler, "RFC 4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks", 2007.

[12] IEEE Standard, 802.15.4-2006, "Wireless medium access control and physical layer specifications for low-rate wireless personal area networks", 2006.

[13] N. Kushalnagar, G. Montenegro and C. Schumacher, "RFC 4919: IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", 2007.

[14] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "RFC 4129: The Kerberos Network Authentication Service (V5)", Request for Comments, 2005.

A Threat Model Approach to Attacks and Countermeasures in On-line Social Networks

Borja Sanz*, Carlos Laorden*, Gonzalo Alvarez† and Pablo G. Bringas*
*Laboratory for Smartness, Semantics and Security (S^{3}Lab), University of Deusto
Bilbao, Spain
Email: {borja.sanz, claorden, pablo.garcia.bringas}@deusto.es
†Instituto de Física Aplicada, Consejo Superior de Investigaciones Científicas (CSIC)
Madrid, Spain
Email: gonzalo@iec.csic.es

***Abstract*—On-line Social Networks (OSN) have become the most popular Internet service today. OSN are being embraced by companies and organisations to help connecting people, assist dealing with cooperative tasks, and develop marketing and public relations campaigns. Despite all their benefits and advantages, as happens with every new technology, they are prone to several security issues. In addition to privacy concerns, there are many other dangerous vulnerabilities that affect security. In this paper, we present our Threat Modelling in OSN, which focuses on identifying attacks against users of OSN and possible countermeasures to mitigate the risks.**

I. Introduction

On-line Social Networks (OSN) have become the most visited sites surpassing information gatherers like Google, MSN, or Yahoo!, consuming most of the time that users spend connected to the Internet, both via desktop and mobile devices.

Although there is no accepted and universal definition for OSN, this paper will use the working definition provided by INTECO and the Agencia Española de Protección de Datos [1]:

'Services that let their users to create a public profile where they can introduce personal data and information. The users have different tools to interact with each other.'

Many enterprises are embracing OSN and integrating them within their strategic plans: viral marketing campaigns; collaborative working environments within the enterprise to allow a free knowledge flow in the new paradigm known as *Enterprise Social Networking* (ESN) [2]; image and reputation promotion of enterprises and people within the enterprises; collaborative content creation via wikis, blogging or microblogging; information exchange with faithful and potential clients, partners, or competitors; search for candidates; etc.

Unfortunately, along with the aforementioned personal and corporative benefits come several web-platform-dependant threats. As expected, with the expansion of OSN, both in and out the enterprise, they are becoming the favourite target for cybercriminals. Actually, in 2009, OSN were one of the main significant channels to identity theft and information leaking [3], [4], [5], [6]. Furthermore, spam sending and malware distribution through OSN are increasing at an incredible pace [7], [8].

The remainder of this paper is organised as follows: Sec. II provides a short introduction to Threat Modelling (TM); Sec. III presents the assets at risk by OSN; Sec. IV details the attacks that are appearing against those assets through OSN; Sec. V discusses some of the countermeasures to be implemented against the previous attacks; finally, Sec. VI concludes and outlines the avenues of future work.

II. Threat Modelling

Threat Modelling is a description of a collection of security aspects, a set of plausible attacks which are able to affect the performance of any computer system. This methodology allows security experts to identify security risks, verify an application's security architecture, and develop countermeasures in the design, coding, and testing phases [9]. Therefore, analysing and modelling the potential threats that an application faces is an important step in the process of designing a secure application [10].

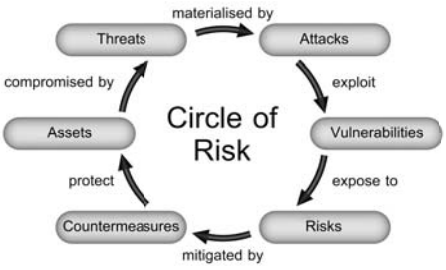


Fig. 1. Threat Modelling's Circle of Risk.

Being the main objective of threat modelling to provide useful guidelines on how to mitigate the associated risks, we must be able to distinguish the elements corresponding to what we have called the *Circle of Risk* (CoR) (shown in Fig. 1). The CoR is composed of *assets*, which are compromised by *threats*; threats that exploit *vulnerabilities*, which when misused result in *exposure*, which represents a serious *risk*. Finally, the *countermeasures* mitigate the dangers caused by those risks; countermeasures which have as goal protecting the assets. Definitions for the aforementioned terms can be found within the technical dictionaries [11] and [12].

Although the threat modelling process requires the study in detail of every above-mentioned element, in this paper we introduce a first approach to the CoR, focussing on the assets, attacks, and countermeasures.

III. Assets at risk by OSN

Every organisation has at disposal several assets that must be protected to guarantee the proper course of its business. The loss, theft, destruction, reduction, or damage of any of these assets could prevent the organisation from achieving its objectives. Therefore, among the assets specially threatened by OSN we can identify [13]:

1) Private information: it can be stolen or used against its legitimate owner in order to harass, extort, or send hyper-contextual advertising.
2) Financial assets: they can be stolen through on-line banking fraud, telephone fraud, or lost by decreased productivity.
3) Intellectual property: it can be stolen, plagiarised, or illegally distributed free of charge, causing economic losses.
4) Corporate secrets: their leakage or theft can cause economic losses, reputation damage, or decreased competitiveness.
5) User's physical security: it can be compromised by stalkers, harassers, criminals, or thieves.
6) Computing and network resources: they can be consumed leading to denial of service or decreased Quality of Service (QoS).
7) Corporate and personal reputation: it can be irreversibly damaged.
8) Digital identity: it can be spoofed or stolen.

In conclusion, the misuse of OSN affects the aforementioned assets, which are compromised by the attacks described in the next section.

IV. Attacks in OSN

OSN have concocted a dangerous cocktail of user-supplied content, open APIs, and web pages heavily loaded with JavaScript and embedded media of all descriptions. And it is an environment that is largely devoid of security standards and practices [14]. Since attacks are aimed at the aforementioned assets, this work introduces the potential attacks that affect OSN organised in categories corresponding to the objective they are oriented to.

A. Private Information

- **Sensitive data retrieval:** Attackers are able to collect users' personal data due to their negligence when publishing private information [15], [16], [17].
- **Sensitive attribute inference models:** The attributes of users who are connected in social networks are often correlated. Zheleva et al. [3] introduced different attacks to infer the hidden sensitive values:
  - ***Friend-Aggregate model* (AGG):** AGG looks at the sensitive attribute distribution amongst the friends of the person under question.
  - ***Collective classification model* (CC):** Unlike more traditional methods, in which each instance is classified independently of the rest, collective classification aims at learning and inferring class labels of linked objects together.
  - ***Flat-link model* (LINK):** Another approach to dealing with links by 'flattening' the data by considering the adjacency matrix of the graph.
  - ***Blockmodelling attack* (BLOCK):** The basic idea behind stochastic blockmodelling is that users form natural clusters or blocks, and their interactions can be explained by the blocks they belong to.
  - ***Groupmate-link model* (CLIQUE):** One can think of groupmates as friends to whom users are implicitly linked. In this model, they assume that each group is a clique of friends, thus creating a friendship link between users who belong to at least one group together.
  - ***Group-based classification model* (GROUP):** Another approach to dealing with groups is to consider each group as a feature in a classifier, inferring sensitive information according the groups a user belongs to.
  - **BASIC** In the absence of relationship and group information, the only available information is the overall marginal distribution for the sensitive attribute in the public profiles. So, the simplest model is to use this as the basis for predicting the sensitive attributes of the private profiles.
- **Data Mining for demographic information:** Using data mining techniques to retrieve public demographic data [18], one could infer unpublished personal data about other users.
- **Automated User Profiling:** Retrieval of users sensitive data by querying social networks for registered e-mail addresses and crawling every profile found to collect personal information [19].
- **De-anonymise OSN users:** It exploits group membership information that is available on social networking sites, which is often sufficient to uniquely identify users, or, at least, to significantly reduce the set of possible candidates [20].
- **OSN Mash-ups:** Link data between independently provided web services to obtain previously unforeseen inferences including highly personal information [21].
- **OSN Aggregators:** Services that integrate several OSN which multiply vulnerabilities by giving read/write access to several social network accounts using a single weak authentication [21].

B. Financial Assets

- ***Cross-Site Scripting* (XSS):** A type of computer security vulnerability typically found in web applications that

enables malicious attackers to inject client-side script into web pages viewed by other users [22], [21], [23].

- ***Cross-Site Request Forgery* (CSRF):** Unlike XSS, which exploits the trust a user has for a particular site, CSRF exploits the trust that a site has in a user's browser [22], [23].
- **Bank-customer oriented Malware:** In order to maximise their monetary benefits, malware creators target bank customers credentials [24]. The appearance of these attacks has increased [25], due to the use of social networks as a distribution channel. A recent example is *Koobface*[1], which upon successful infection, gathers sensitive information from the victims such as credit card numbers.

C. Intellectual Property

- **Contents publication property of third parties:** It occurs when a user publishes the contents not being the legitimate holder of the intellectual property rights of such material [1].
- **Search engines indexation of protected contents** entails a greater diffusion and therefore an exponential increased number of reproductions [1].
- **Loss of control over contents when users unsubscribe from the on-line service:** OSN based on profiles eliminate, or at least block, all the contents associated to the profile of the user leaving the service, while in platforms based on contents, where members can get to publish works without being associated directly to their profile, material may remain publicly accessible [1].

D. Corporate Secrets

- **Social Engineering:** Manipulating people into performing actions or divulging confidential information using information found in OSN profiles [26].
- **Spear Phishing:** Spear phishing appears genuine to all the employees or members within a certain company, government agency, organisation, or group, using information found in OSN profiles [27].

E. Physical Security

- **Location Inferring** from recognisable places in the image [28] or connection IP [29].
- **Facial Recognition:** Sophisticated facial recognition algorithms used to identify unknown users [30].
- **Harassment between Adults** Bullying via electronic communication tools [31], [32].
- **Cyber-bullying** Harassment via electronic communication tools from child to child [31], [32].
- **Cyber-grooming (harassment from adult to child)** Sexual exploitation of children on-line [33].

[1]http://news.cnet.com/koobface-virus-hits-facebook/

F. Computing and Network Resources

- **Spam and Hyper-contextualised Advertising:** Spam is becoming a major issue for OSN, and the use of hyper-contextualised advertising (i.e. adapt advertising to users preferences) increases the possibility of the junk messages being read [21].
- **Botnets:** Attacks designed solely to disable infrastructure to those that also target people and organisations.[34].

G. Corporate and Personal Reputation

- **Sybil Attacks:** Given a reputation system, a peer may attempt to falsely raise its reputation by creating fake identities – or sybils – and using them to its benefits [35].
- **Classes of attacks against reputations systems:** Hoffman et al. [36] classify attacks against reputation systems based on the goals of the reputation systems.
  - **Self-promoting:** Attackers manipulate their own reputation by falsely increasing it.
  - **Self-Serving or Whitewashing:** Attackers escape the consequence of abusing the system by using some system vulnerability to repair their reputation. Once they restore their reputation, the attackers can continue the malicious behaviour.
  - **Slandering:** Attackers manipulate the reputation of other nodes by reporting false data to lower their reputation.
  - **Orchestrated:** Attackers orchestrate their efforts and employ several of the above strategies.
  - **Denial of Service (DoS):** Attackers may cause denial of service by either lowering the reputation of victim nodes so they cannot use the system or by preventing the calculation and dissemination of reputation values.

H. Digital Identity

- **Credentials Theft** by technical hacking techniques [37].
- **Profile Cloning** consists of identifying a victim and creating a new account with his real name and photograph inside the same social network [38].
- **Cross-site Profile Cloning** identifies victims who are registered in one social network, but not in another and steals their identities creating accounts for them in the network where they are not registered [38].

Finally, it is important to take into account that the danger level of one attack is directly proportional to how dangerous the vulnerability being exploited is and inversely proportional to the effectiveness of the deployed countermeasures.

V. Major Countermeasures in OSN

Countermeasures reduce the vulnerabilities in a system. In this section, we present these countermeasures grouped into the following main categories: platform countermeasures and user countermeasures. The former refers to countermeasures which collaborative platforms must implement in order to prevent attacks directed both to platforms and users, while the

later intends to introduce the best practices to improve users privacy habits.

A. Countermeasures addressed to the Platform

1) Technological Security of the Platform: System administrators of collaborative networks should be aware that their users share personal data through their services. Therefore, they should protect their networks against potential attacks, employing tools especially made to combat against pharming and phishing [39] cases, not to mention one of the most annoying threats of the current times: the spam[2]. Regarding network connections, they should make use of secure connections applying technologies (e.g. *Security Socket Layer* (SSL) [40]), to ensure private data transmissions.

On the other hand, OSN provide users with little control over their personal data. As a consequence, identity theft and fake profiles are common issues. These platforms should provide tools to prevent cases of identity theft, to allow legitimate users to get back the control of the account after the theft, or to strengthen user identification before registration. Additionally, it is recommended to implement technological measures to verify the age of the users, in order to protect children against inappropriate contents or behaviours.

2) User's Data: OSN need to facilitate access to the Terms of Service and User Conditions displaying all the information in understandable terms. To this end, these documents must employ a perfectly understandable language by any kind of user. After reading the document, the user should know its rights and obligations during the use of the service.

Besides, OSN must guarantee the users a complete control over their published information. Therefore, a social network must implement several procedures in order to satisfy the following:

- Users should know the intended use by the social network of both personal and published data.
- Users should be able to apply the rights to access, rectify, cancel, and oppose to data concerning them published in the OSN.
- User profile configuration should default to maximum privacy, allowing to later changing it according to personal preferences.
- Users should be able to prevent the publication of unauthorised data. The use of tagging mechanisms requesting user's approval is one of the approaches aimed at the achievement of this goal.

Furthermore, OSN must protect users data against the indexation of search engines by using appropriate codification.

3) Author's royalties protection: Author's rights must be protected. OSN must provide users with tools that allow reporting the existence of contents protected by author's rights. Additionally, social networks need adequate staff or automatic tools, such as Digital Right Management tools, anti-copy systems, or watermarks, to check all uploaded contents and establish if such contents are subject to intellectual rights.

[2]http://blog.facebook.com/blog.php?post=40218392130

Besides, OSN users must know the nature of the rights to authorship and the importance to respect them for the correct use of the service, through general conditions when creating new accounts, FAQs, etc.

4) User Awareness: Users should be informed about the use that social networks make of their personal data, the advertisement systems present in the platform, and the potential threats that users face while using on-line services. Similarly, it is necessary to display information related to the security of the platform, including the measures that users should take in case of abuse of their rights.

B. Countermeasures addressed to the Users

1) User's Behaviour: The user must read the Terms of Use and Privacy Policies of the OSN, both before the registering process and every time any change occurs. Once the user has registered, it must configure properly the privacy settings, so that only his friends have access to the published contents.

Users have absolute control over the information that they want to publish inside the OSN. They are therefore responsible for the publication of excessive information putting at risk their intimacy or their whereabouts. In this sense, it is mandatory to raise the users awareness into a rational publishing of their personal information. However, users should be aware that some information is out of their control, for example when published by other users of the same OSN, by public organisms, such as *Boletín Oficial del Estado* (BOE), or public registries.

Moreover, friendship relations are the core of these networks. Once defined the privacy settings, users must be careful with friend requests. Users should only accept friend requests coming from people already known and avoid accepting compulsively any request for friendship because it could result in privacy issues.

2) Technological Concerns: There are security and technological considerations that users must take into account in order to increase the level of security. First, users should use different user-names and passwords to access different social networks. Second, they should use strong passwords to prevent brute force attacks. Finally, they should use updated security software and operating system.

3) Special Considerations for Children: Under-age users are specially vulnerable. Thus, they need extra care to ensure that their personal data is not disclosed. Parents or guardians should be consulted for every sensitive action when using social networks (e.g. content uploading and publishing personal information), being able to abort their children actions.

Additionally, parents and guardians should take into account several considerations. The computer should be placed in a common area of the house, establishing some rules about the use of Internet. Parental control and content-blocking systems should be installed and effectively working, and minors should be aware of the dangers that OSN might represent.

VI. Conclusion

On-line Social Networks represent one of the last and most important Internet services. Albeit most enterprises hesitate

whether to ignore completely the OSN, this new phenomenon can not be ignored, but neither can be integrated into the business model without knowing the risks. In this paper, we presented a first approach to an OSN Threat Modelling that discovers the first elements to take into account when attempting to protect a system. To that end, we identify the assets at risk, the attacks that can compromise them, and we propose some countermeasures to protect against these attacks (the mapping attack-countermeasure is provided in Table I). In this table we can appreciate that the number of different attacks outnumbers the countermeasures, which indicates that a good level of security can be assured by following some simple guidelines. In addition, the analysis of the attacks and countermeasures shows that user's good practice is the main objective in order to achieve more secure OSN.

The future work of this OSN TM is oriented in three main directions. First, we will complete the aforementioned 'Circle of Risk' (see Fig. 1), with the exposures that suffer the assets and the risks that represent them. Second, we plan on developing a taxonomy which organises all the existing OSN threats, attacks, vulnerabilities, and countermeasures. Finally, we will study the feasibility of adding weighted variables to the taxonomy in order to help identifying assets at risk and, hence, supporting the hardening of a system.

ACKNOWLEDGEMENTS

The work described in this paper was supported by the Spanish *Ministerio de Ciencia e Innovación*, project CUCO (MTM2008-02194), and *Ministerio de Industria*, project Cenit SEGUR@, Security and Trust in the Information Society, (BOE 35, 09/02/2007, CDTI).

REFERENCES

[1] Study on the privacy of personal data and on the security of information in social networks, Tech. rep., INTECO (2009).
[2] J. DiMicco, D. Millen, W. Geyer, C. Dugan, B. Brownholtz, M. Muller, Motivations for social networking at work, in: Proceedings of the ACM 2008 conference on Computer supported cooperative work, ACM New York, NY, USA, 2008, pp. 711–720.
[3] E. Zheleva, L. Getoor, To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles, in: Proceedings of the 18th international conference on World wide web, ACM New York, NY, USA, 2009, pp. 531–540.
[4] B. Krishnamurthy, C. Wills, On the leakage of personally identifiable information via online social networks, in: Proceedings of the 2nd ACM workshop on Online social networks, ACM, 2009, pp. 7–12.
[5] J. Lindamood, R. Heatherly, M. Kantarcioglu, B. Thuraisingham, Inferring private information using social network data, in: Proceedings of the 18th international conference on World wide web, ACM, 2009, pp. 1145–1146.
[6] B. Chen, D. Kifer, K. LeFevre, A. Machanavajjhala, Privacy-Preserving Data Publishing, Foundations and Trends in Databases 2 (1-2) (2009) 1–167.
[7] Z. Mazur, H. Mazur, T. Mendyk-Krajewska, Security of Internet Transactions, Internet-Technical Development and Applications (2009) 243.
[8] W. Luo, J. Liu, J. Liu, C. Fan, An analysis of security in social networks, Dependable, Autonomic and Secure Computing, IEEE International Symposium on 0 (2009) 648–651.
[9] F. Swiderski, W. Snyder, Threat modeling, Microsoft Press Redmond, WA, USA, 2004.
[10] L. Desmet, B. Jacobs, F. Piessens, W. Joosen, Threat modelling for web services based web applications, in: Eighth IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS 2004), Springer, 2004, pp. 161–174.
[11] U. E. Gattiker, The Information Security Dictionary: Defining The Terms That Define Security For E-business, Internet, Information And Wireless Technology (KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE), Kluwer Academic Publishers, Norwell, MA, USA, 2004.
[12] R. Slade, Dictionary of Information Security, Syngress Media Inc, 2006.
[13] G. Alvarez, Amenazas 2.0: los riesgos de las redes sociales online en la empresa, PC World 274 (2010) 64–70.
[14] S. Mansfield-Devine, Anti-social networking: exploiting the trusting environment of Web 2.0, Network Security 2008 (11) (2008) 4–7.
[15] B. Ng, A. Kankanhalli, Y. Xu, Studying users' computer security behavior: A health belief perspective, Decision Support Systems 46 (4) (2009) 815–825.
[16] J. Bryce, M. Klang, Young people, disclosure of personal information and online privacy: Control, choice and consequences, Information Security Technical Report 14 (3) (2009) 160–166.
[17] B. Huberman, E. Adar, L. Fine, Valuating privacy, IEEE security & privacy 3 (5) (2005) 22–25.
[18] D. Jensen, J. Neville, Data mining in social networks, in: Dynamic Social Network Modeling and Analysis: workshop summary and papers, 2003, pp. 287–302.
[19] M. Balduzzi, C. Platzer, T. Holz, E. Kirda, D. Balzarotti, C. Kruegel, Abusing social networks for automated user profiling, Tech. Rep. EURECOM+3042, Institut Eurecom, France (03 2010).
[20] G. Wondracek, T. Holz, E. Kirda, S. Antipolis, C. Kruegel, A Practical Attack to De-Anonymize Social Network Users.
[21] G. Hogben, Security issues and recommendations for online social networks, Position Paper. ENISA, European Network and Information Security Agency.
[22] G. Alvarez, S. Petrovic, A new taxonomy of web attacks suitable for efficient encoding, Computers and Security 22 (5) (2003) 435–449.
[23] Web security threat classification v2.0, Tech. rep., Web Application Security Consortium (2010).
URL `http://projects.webappsec.org/f/WASC-TC-v2_0.pdf`
[24] M. Pemble, Evolutionary trends in bank customer-targeted malware, Network Security 2005 (10) (2005) 4–7.
[25] P. Gutmann, The commercial malware industry, in: DEFCON conference, 2007.
[26] M. Workman, Gaining access with social engineering: An empirical study of the threat, Inf. Sys. Sec. 16 (6) (2007) 315–331.
[27] T. N. Jagatic, N. A. Johnson, M. Jakobsson, F. Menczer, Social phishing, Commun. ACM 50 (10) (2007) 94–100.
[28] M. Zhang, Content-based Image retrieval, Artificial Intelligence for Maximizing Content Based Image Retrieval (2009) 115.
[29] Y. Jiang, B. Fang, M. Hu, X. Cui, Techniques for determining the geographic location of IP addresses in ISP topology measurement, Journal of Computer Science and Technology 20 (5) (2005) 689–701.
[30] P. Phillips, Support vector machines applied to face recognition, Advances in Neural Information Processing Systems (1999) 803–809.
[31] T. Beran, Q. Li, Cyber-harassment: A study of a new method for an old behavior, Journal of Educational Computing Research 32 (3) (2005) 265–277.
[32] Q. Li, Cyberbullying in schools: A research of gender differences, School Psychology International 27 (2) (2006) 157.
[33] D. Roberts, Cyber-Victimisation in Australia: Extent, Impact on Individuals and Responses (2008).
[34] E. Cooke, F. Jahanian, D. McPherson, The zombie roundup: Understanding, detecting, and disrupting botnets, in: Proceedings of the USENIX SRUTI Workshop, 2005, pp. 39–44.
[35] A. Cheng, E. Friedman, Sybilproof reputation mechanisms, in: P2PECON '05: Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems, ACM, New York, NY, USA, 2005, pp. 128–132.
[36] K. Hoffman, D. Zage, C. Nita-Rotaru, A survey of attack and defense techniques for reputation systems, ACM Comput. Surv. 42 (1) (2009) 1–31.
[37] J. Erickson, Hacking: the art of exploitation, 2nd edition, No Starch Press, San Francisco, CA, USA, 2008.
[38] L. Bilge, T. Strufe, D. Balzarotti, E. Kirda, All your contacts are belong to us: automated identity theft attacks on social networks, in: Proceedings of the 18th international conference on World wide web, ACM New York, NY, USA, 2009, pp. 551–560.
[39] M. Jakobsson, S. Myers, Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft, Wiley-Interscience, 2006.
[40] E. Rescorla, SSL and TLS: designing and building secure systems, Addison-Wesley, 2001.

TABLE I
RELATIONSHIP BETWEEN ATTACKS AND COUNTERMEASURES.

Attacks \ Countermeasures	User control of published information	Only accept friend request from previously known people	Strength user identification system	Use different user-names and passwords	Updated platform and user's security software	Author's rights surveillance tools	Inform users and authors about intellectual rights	Appropriate HTML codification and robot.txt configuration	Inform users about Terms of Use in Social Network	User education about security concerns
Sensitive data retrieval	✓	✓								
AGG	✓	✓								
CC	✓	✓								
LINK	✓	✓								
BLOCK	✓	✓								
CLIQUE	✓	✓								
GROUP	✓	✓								
BASIC	✓	✓								
Data mining demographic information	✓	✓								
De-anonymise OSN users	✓		✓	✓						
OSN agregators	✓									
OSN Mash-ups	✓									
Automated User Profiling	✓		✓	✓						
Cross-Site Scripting					✓					
Cross-Site Request Forgery					✓					
Bank-customer oriented malware					✓					
Third party contents publication						✓	✓			
Indexation of protected contents								✓		
Control loss over contents									✓	
Social engineer		✓								✓
Spear phishing					✓					✓
Location inferring from recognisable places in image	✓									✓
Location inferring from connection IP										✓
Facial recognition	✓									✓
Harassment between adults										✓
Cyber-bullying										✓
Cyber-grooming										
Spam and Hyper-contextualised advertising					✓					✓
Botnets					✓					
Sybil attacks										✓
Self-promoting										✓
Self-Serving or Whitewashing										✓
Slandering:										✓
Orchestrated										✓
Denial of Service										✓
Credentials theft			✓	✓						
Profile cloning			✓	✓						
Cross-site profile cloning			✓	✓						

Distribución segura de componentes software basada en OpenID

Isaac Agudo, Jose A. Onieva, Daniel Merida
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga
Email: {isaac,onieva,dmerida}@lcc.uma.es

***Resumen*—En la actualidad, cada vez son más frecuentes los ataques software mediante la utilización de malware o sustitución de programas (o componentes) en los repositorios a los cuales los usuarios finales (o máquinas) acceden. Esta situación se ve de alguna manera acentuada con el dinamismo existente en la programación y ejecución de estos componentes, en la que distintos desarrolladores pueden participar para desplegar un determinado servicio o parte de él.**

Por ello, en este artículo se presenta una solución para la distribución de código de forma segura usando OpenID y firmas con certificados de clave pública de corta duración. De esta forma, se consigue un compromiso de seguridad que permite distribuir código firmado sin la necesidad de que los desarrolladores dispongan a priori de un certificado específico. Presentamos además algunos detalles acerca de la implementación realizada para hacer realidad este diseño.

I. INTRODUCCIÓN Y OBJETIVOS

La distribución segura de componentes software es en la actualidad un factor crítico para todos los usuarios, y no sólo para las empresas desarrolladoras. Cada vez son más frecuentes los ataques software mediante la utilización de malware o sustitución de programas (o componentes software) en los repositorios a los cuales los usuarios finales (o máquinas) acceden. Esta situación se ve de alguna manera acentuada por el dinamismo intrínseco en la programación y ejecución de estos componentes software, en la que distintos desarrolladores pueden participar para desplegar un determinado servicio (o partes de él).

Es muy conocido el imparable aumento del malware en todo el mundo. En especial en nuestro país que, según el informe del primer trimestre de este año 2010 de PandaLabs [1], ocupa la primera posición en el ranking de países con respecto al número de ordenadores infectados. Según estas estadísticas, al menos uno de cada tres ordenadores están infectados por algún tipo de malware. Por ello es especialmente importante bloquear en todo momento la capacidad de propagación del mismo.

Dado el aumento de prestaciones de los dispositivos móviles, existe una aplicación disponible para prácticamente cualquier necesidad, y aquí la integridad del software o la confianza en el desarrollador son objetivos fundamentales a perseguir. La proliferación de aplicaciones disponibles para todo tipo de dispositivos y para los dispositivos móviles en particular, ha provocado por añadidura que sean cada vez más el numero de desarrolladores que distribuyen estas aplicaciones. No todos ellos deben tener el mismo nivel de confianza sino queremos que el malware encuentre una puerta de entrada a nuestros sistemas.

Este escenario se vuelve más importante cuando pensamos en el futuro: Multitud de dispositivos, posiblemente autónomos, interconectados entre sí y conectados al "exterior", siendo así actores del *Internet of Things*, y ejecutando servicios cuyo código y composición puede cambiar de forma dinámica, y en los que la instalación y desinstalación de los distintos componentes que construyen el servicio en ejecución se ha de realizar en tiempo real.

Existen soluciones que permiten decir a nuestro sistema que sólo ejecute componentes firmados y que además esta firma sea confiable (nuestro sistema ha de confiar en alguno de los certificados existentes en el camino de verificación de la firma). Pero no escapa a los usuarios finales (ni a los desarrolladores) que estas soluciones transforman a los sistemas en ocasiones en demasiados restrictivos, haciendo que en muchos casos, los administradores de estos dispositivos terminen por cancelar este tipo de comprobaciones.

Por otra parte, los desarrolladores, se ven abocados a soportar una burocracia digital, para en todo momento, contar con certificados digitales no caducados y firmados por una entidad confiable. Este proceso es relativamente sencillo y natural para grandes empresas desarrolladoras de software. Sin embargo, no ocurre lo mismo para desarrolladores particulares que carecen de medios. En los ambientes móviles el uso de código firmado está muy extendido y aunque algunas plataformas como Symbian proporcionan mecanismos para la creación de certificados de desarrollo [2], este proceso requiere de un registro previo del desarrollador y la validez del código firmado está restringida al teléfono de desarrollo registrado.

Esta situación se agrava en entornos en los que los servicios (programas en ejecución) se componen de forma dinámica y en tiempo real de distintos módulos o componentes, tal y como ocurre en la plataforma OSGi [4]. La plataforma OSGi es un sistema de módulos y servicios para el lenguaje de programación Java que implementa un modelo completo y dinámico de componentes. Las aplicaciones o componentes (en forma de *bundles*) pueden ser instaladas de forma remota, iniciadas, detenidas, actualizadas y desinstaladas sin necesidad de reiniciar. La gestión del ciclo de vida se realiza a través de APIs que permiten la descarga remota de políticas de gestión. El Registro de Servicios permite a los bundles detectar la

adición o la supresión de los servicios, y adaptarse en consecuencia. Las especificaciones OSGi se han movido más allá del enfoque original, y ahora se utilizan en aplicaciones que van desde teléfonos móviles al IDE de Eclipse [5], pasando por un amplio espectro de áreas de aplicación tan heterogéneas como pueden ser los automóviles, la automatización industrial y la automatización de edificios, dispositivos móviles como las PDA's, el ocio y el entretenimiento, la gestión de flotas o los servidores de aplicaciones.

Así, el escenario que se presenta (ver Figura 1 para más detalles) es el de un dispositivo que soporta OSGi que con el objeto de lograr la ejecución dinámica de componentes cuenta con conexión constante a distintos *repositorios* OSGi de los que cargará los distintos bundles de acuerdo a las dependencias y necesidades de los bundles actuales en ejecución. Nuestra solución ha de tener en cuenta los puntos de vista y necesidades de este dispositivo y de los desarrolladores para proporcionar integridad y confianza a la hora de ejecutar estos componentes a la par que facilidad de uso y sencillez a la hora de subir estos componentes a los repositorios.

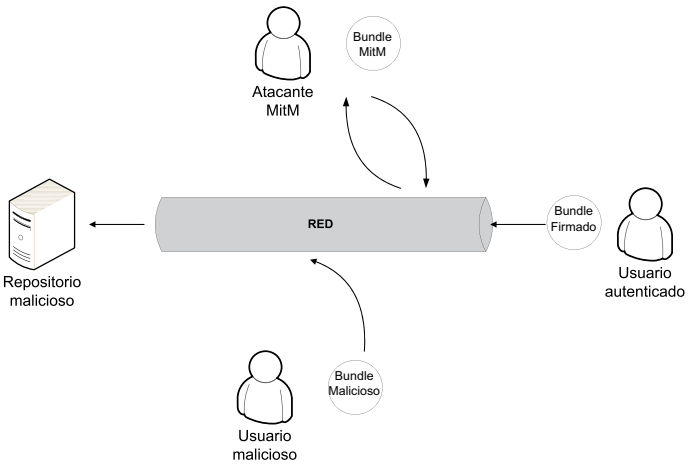


Figura 1. Amenazas en el proceso de desarrollo de bundles

II. REQUISITOS

Las amenazas a la seguridad para el despliegue de los bundles [6] pueden ser de tres tipos:

1. La presencia de repositorios maliciosos para la publicación de bundles.
2. Ataques "Man in the Middle", de forma que un atacante pueda modificar un bundle o sustituirlo completamente por otro durante la carga o la descarga.
3. La posibilidad de que un atacante acceda al repositorio de bundles o a la plataforma cliente para modificar los componentes almacenados en ellos.

Tras el análisis del escenario y sus principales amenazas, los principales requisitos que pueden extraerse del mismo son los siguientes:

1. Sencillez.
2. Movilidad / portabilidad.
3. Autenticación del desarrollador.
4. Integridad de los bundles.
5. Generación de firmas seguras.

Se pretende conseguir una plataforma **sencilla** y robusta para la firma de componentes software. El impacto para el desarrollador (en términos de tiempo y complejidad) que sube un bundle a un repositorio OSGi ha de ser mínimo, sin que ello dificulte el proceso de verificación por parte de los usuarios[1] o dispositivos.

Al no requerirse el uso de certificados personales para la distribución de los bundles se permite cierta **movilidad** de los desarrolladores, ya que pueden subir los bundles a los repositorios OSGi simplemente accediendo al repositorio con su usuario y contraseña habitual.

Es importante considerar que en el modelo de desarrollo de software libre, el conjunto de desarrolladores es dinámico y que la implicación de estos en el proyecto es variable. Por tanto se tiene que dar cabida no solo a los desarrolladores estables sino también a los eventuales que solo quieren aportar su granito de arena al proyecto. Por este motivo se define en mecanismo de autenticación para el repositorio basado en federación de identidades, de forma que el repositorio no tenga que **autenticar** directamente al desarrollador sino que sea su proveedor de identidad el que proporcione la información necesaria.

En la solución propuesta se utilizan certificados temporales (de vida corta) generados a partir del conjunto de atributos que definen la identidad de desarrollador proporcionado por el proveedor de identidad para firma el código y por tanto proporcionar **integridad** de éste. Además, al ser certificados temporales, pueden ser utilizados desde cualquier dispositivo independiente de su situación geográfica y evitando la implicación en seguridad que tendría llevar consigo siempre un par de claves pública y privada para la firma de estos componentes.

El uso de una federación de identidad permite que el desarrollador solo tenga que autenticarse de la misma manera (y con los mismos credenciales) que lo hace en otros servicios para poder almacenar un bundle firmado. Por supuesto, el repositorio OSGi cuenta con políticas de autorización que determinarán la visibilidad de estos bundles para otros usuarios, aunque consideramos este proceso fuera del ámbito de este trabajo.

Para la **generación de firmas de forma segura** por parte del usuario, esta plataforma permitirá al usuario generar el par de claves de forma local y será el repositorio OSGi quien haga la función de tercera parte confiable que certifique esas claves, y que certifique que su usuario legítimo es quien dice ser. Este esquema difiere del sistema de distribución de paquetes de Debian [3], donde los repositorios pueden firmar sus paquetes permitiendo su posterior verificación, ya que en nuestro esquema para la verificación de un bundle el filtro de confianza se hace a nivel de desarrollador y no de repositorio.

[1]Estos usuarios, potencialmente, pueden ser a su vez desarrolladores que trabajen de manera colaborativa en la implementación de bundles.

III. COMPONENTES DE LA PLATAFORMA

III-A. Bundles

Un bundle no es más que un fichero JAR (Java Archive) que contiene además un conjunto de metadatos que especifican las características del componente. Estos metadatos están incluidos dentro del archivo `MANIFEST.MF`. Por tanto:

Bundle = archivo `JAR` + `MANIFEST.MF` con metadatos

Es necesario proporcionar algún tipo de soporte para el ciclo de vida de los bundles, desde la instalación hasta la ejecución y el borrado, por lo que la seguridad deberá contemplarse a lo largo de todo el ciclo de vida.

III-B. Repositorio OSGi

Se trata del repositorio en el que se almacenan los distintos bundles al que las máquinas que cuenten con plataformas de ejecución OSGI acceden para descargar componentes. De la misma manera, los desarrolladores suben estos bundles una vez testeados o con el objetivo de colaborar en el desarrollo de los mismos.

El administrador del repositorio ha de autenticar a todos los desarrolladores que pretendan almacenar bundles y a su vez facilitar la integridad de estos componentes, de forma que se enlace de manera inequívoca cada bundle con cada uno de sus autores. Para ello, la plataforma proporciona un servicio o autoridad de certificación (CA) cuyo único objetivo es permitir la firma de los mismos.

Ciertamente, este servicio de certificación podría ser externo, y reconocido, pero dado que los certificados han de ser temporales y de una funcionalidad muy reducida, el repositorio puede ejercer esta función, reduciendo así el proceso previo a la firma por parte de los desarrolladores.

III-C. Servidor OpenID

OpenID [7] es un sistema de autenticación digital descentralizado, con el que un usuario puede identificarse en una página Web a través de una URL (o un XRI en la versión actual) y puede ser verificado por cualquier servidor que soporte el protocolo. En los sitios que soporten OpenID, los usuarios no tienen que crearse una nueva cuenta de usuario para obtener acceso. En su lugar, solo necesitan disponer de un identificador creado en un servidor OpenID (OpenID provider), es decir, un proveedor de identidad OpenID (IdP).

Este proveedor de identidad puede confirmar la identificación OpenID del usuario a un sitio que soporte este sistema. A diferencia de arquitecturas SSO (Single Sign-On), OpenID no especifica el mecanismo de autenticación. Por lo tanto, la seguridad de una conexión OpenID depende de la confianza que tenga el cliente OpenID en el proveedor de identidad. Si no existe confianza en el proveedor, la autenticación no será adecuada para servicios bancarios o transacciones de comercio electrónico. Sin embargo, el proveedor de identidad puede usar autenticación fuerte si el proveedor de servicio (Service Provider) así lo requiere.

De entre los atributos que proporciona OpenID (ver [7] para más detalles), se han seleccionado los siguientes para la generación de los certificados temporales:

1. openid.ext1.value.firstname (en adelante, firstname)
2. openid.ext1.value.lastname (en adelante, lastname)
3. openid.ext1.value.email (en adelante, e-mail)
4. openid.op_endpoint (en adelante, ID provider)

Con la elección de OpenID, se pretende evitar la centralización que supone un sistema jerárquico de PKI, que restaría dinamismo a la plataforma e introducirían una dependencia sobre la disponibilidad de la jerarquía de CA para la generación de certificados. En otras palabras, logramos con ello evitar la necesidad de acudir presencialmente a una Autoridad de Registro.

III-D. Applet de firma

Para la firma y creación de los certificados de firma digital se requiere la descarga y ejecución por parte del desarrollador de del applet de firma. El applet de firma solo toma como entrada del usuario el repositorio de destino, así como el bundle que se quiere firmar.

La estructura de clases [9] que presenta el código fuente del applet es la que se muestra en la Figura 2

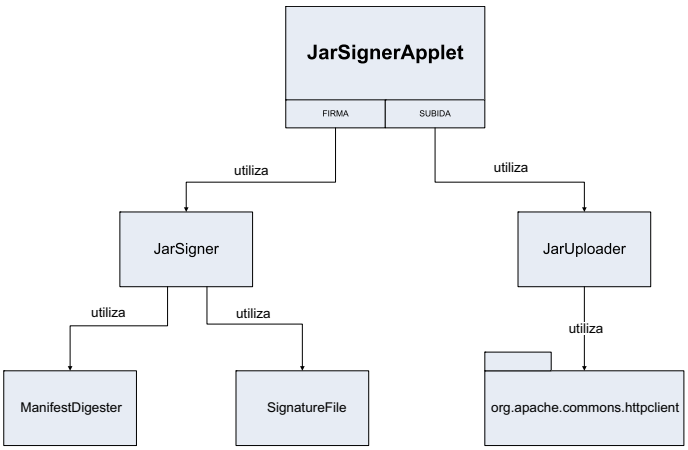


Figura 2. Estructura de clases que utiliza el applet de firma

Como apoyo auxiliar para la firma de bundles se han utilizado dos herramientas proporcionadas por Java: Jarsigner y Keytool.

La herramienta Jarsigner [10] tiene dos objetivos principales: firmar archivos JAR y verificar la firma y la integridad de archivos JAR firmados. Para ello, utiliza información de certificados y claves que obtiene a partir de un almacén de claves (keystore). Un keystore contiene una de base de datos con claves privadas y las cadenas de certificados X.509 que autentican sus correspondientes claves públicas.

La herramienta Keytool [11] se encarga de la gestión de claves y certificados, de forma que permite a los usuarios administrar sus propias claves públicas y privadas y los certificados asociados a éstas para autenticarse a sí mismo frente a otros usuarios o servicios o para realizar verificaciones de integridad mediante las firmas digitales.

Un usuario se puede encontrar con dos tipos diferentes de entradas que formarán parte de un almacén de claves:

1. Claves: Una clave almacenada suele ser una clave privada acompañada de la correspondiente cadena de certificación de su clave pública
2. Certificados de confianza: Los certificados de confianza son aquellos certificados de clave pública de otro usuario o entidad de los que se conoce con certeza la identidad de su propietario

Existen otras opciones de Keytool para importar en el keystore certificados ya existentes, para exportar certificados, para generar certificados autofirmados o para generar solicitudes CSR que serán firmadas posteriormente por una CA, opción ésta de la hemos hecho uso en la implementación.

IV. PROCESOS DE LA PLATAFORMA

El objetivo final de esta plataforma es la distribución segura, mediante un repositorio, de componentes software. Para ello, se utilizará la información de identidad proporcionada por un servidor OpenID[2] para crear unos certificados temporales, que serán firmados por un servicio de CA online del propio repositorio y serán utilizados para la firma del bundle, como paso previo a la carga en el repositorio.

IV-A. Proceso de firma de un bundle

La solución planteada para proporcionar seguridad en los bundles es la firma digital, que por las peculiares características de los bundles, se debe almacenar dentro del bundle, junto al resto de componentes. Los archivos relacionados con la firma de un bundle son los siguientes:

1. Un archivo de manifiesto (Manifest File), que contiene un listado con el valor hash de cada uno de los recursos del bundle.
2. Para soportar varias firmas, la firma digital no se aplica directamente sobre el archivo de manifiesto, sino sobre un archivo para la firma denominado (Signature File) que contiene un valor hash del archivo de manifiesto. Hay un archivo de firmas por cada firmante.
3. La firma digital del archivo de firmas se almacena en un archivo CMS (Content Management System) denominado Block Signature File. La extensión de este archivo es el nombre del algoritmo de firma (.dsa, .rsa, entre otros). Además, este archivo contiene los datos necesarios para la verificación de la firma, como es la cadena de certificados que verifican al usuario firmante, que en nuestro caso será la cadena formada por el certificado temporal creado con los atributos OpenID del usuario y el certificado de clave pública de la CA online del repositorio OSGi.

En nuestra implementación, el orden de estos recursos dentro del bundle es el que aparece descrito anteriormente. Todos estos ficheros se colocarán delante del resto de recursos del bundle. Este orden será uno de los aspectos que se tendrán en cuenta durante el proceso de verificación de un bundle firmado.

[2]En nuestras pruebas se han utilizado con éxito, los servidores OpenID de Google, RedIRIS y Yahoo.

En la Figura 3 se muestra la interacción entre los componentes de la plataforma. El flujo de información en el proceso de firma de un bundle paso por paso sería el siguiente:

- El desarrollador se conecta al servicio de subida de bundles del repositorio (Bundle Upload Service).
- El primer paso será acceder a la URL de descarga del applet. Al acceder a esta URL se comprueba si el usuario accede con atributos OpenID y, en caso de no llevarlos, es redirigido a un servicio WAYF (Where Are You From).
- El servicio WAYF permitirá al usuario seleccionar su proveedor de identidad OpenID, hacia el que será redirigido.
- El usuario introducirá su usuario y contraseña para autenticarse en el servidor OpenID, lo que permitirá que éste servidor envíe los atributos OpenID del usuario hacia la página de redirección, que será la URL de descarga del applet.
- En un segundo acceso a la URL de descarga del applet, si la comprobación de atributos OpenID se realiza con éxito se permite que se inicie el proceso de descarga. En el navegador del usuario, se pedirá la autorización del usuario para la ejecución del applet a partir de la confianza en el certificado del desarrollador del mismo.
- A partir de los atributos OpenID con los que se inicializará el applet, se generan un par de claves pública y privada y la correspondiente solicitud CSR (Certificate Signing Request). Los campos necesarios para las claves se cumplimentarán de la siguiente manera:
  - Common Name: firstname + lastname
  - Organization Unit: e-mail
  - Organization: ID provider
  - Country: ES
- El CSR se envía al servicio de la CA online del repositorio.
- El servicio de la CA online generará un certificado de clave pública a partir del CSR y la clave privada de la CA.
- La CA online envía el certificado de clave pública, junto al certificado de clave pública de la CA si fuera necesario. En el applet, se unen el certificado de clave pública y la clave privada y el usuario ya tiene la capacidad para firmar componentes software con un certificado temporal completo.
- Se realiza el proceso de firma del bundle a partir del certificado temporal obtenido y se envía el bundle al repositorio.

La principal razón para la elección de un applet como elemento central de la plataforma en el proceso de firma es que el par de claves pública y privada se deben generar de forma local en el equipo del usuario (para que el desarrollador realmente confíe en las claves generadas), pero a partir de un software existente en la plataforma.

De esta forma, el usuario se descarga el applet en el navegador y autoriza su ejecución (confiando en su autor) durante el proceso automático de descarga del mismo. La se-

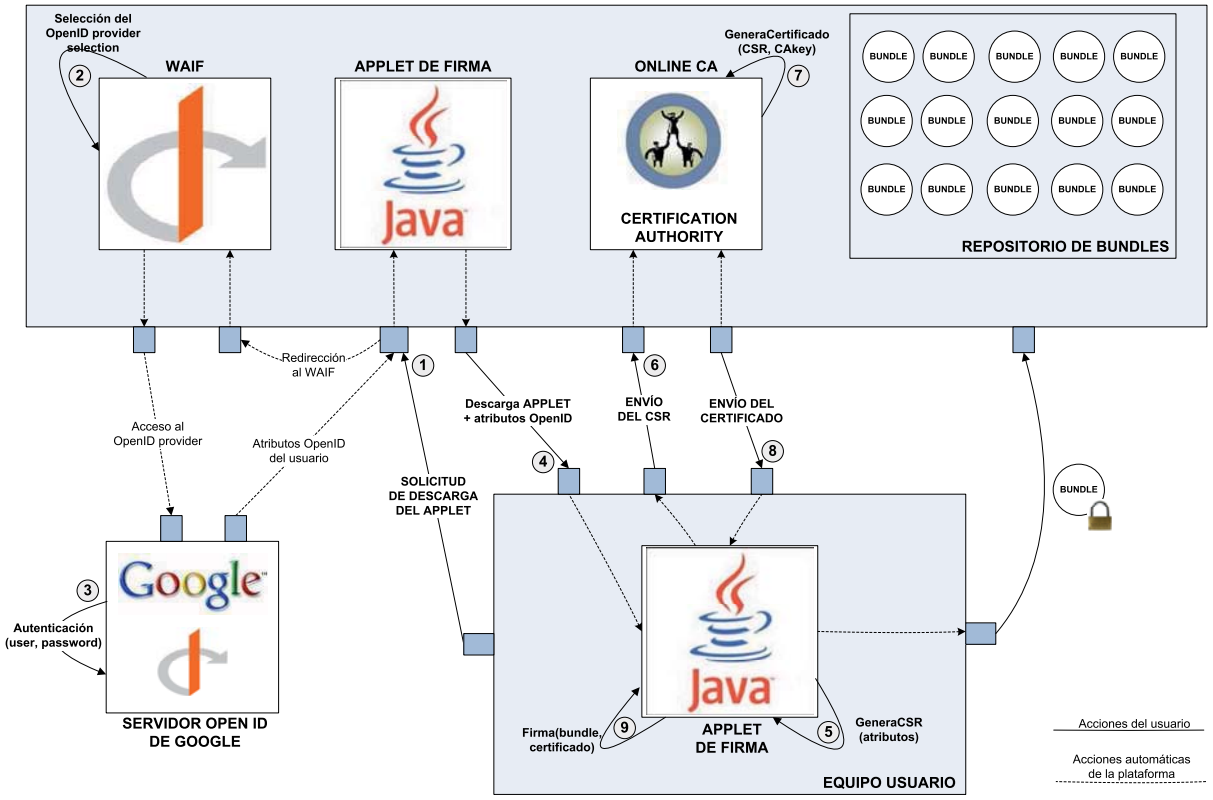


Figura 3. Arquitectura de la plataforma

guridad de este applet y como el usuario deposita su confianza autorizando su ejecución es un problema resuelto que por lo tanto queda fuera del ámbito de este trabajo.

IV-B. Proceso de verificación de un bundle firmado

El proceso de verificación se realiza de forma simétrica al proceso de firma consta de los siguientes pasos:

1. El primer paso será validar la identidad del firmante, para lo que habrá que verificar la cadena de confianza del certificado, que sólo estará formada por el certificado de clave pública de la CA online del repositorio, verificando también que dicha CA se encuentra en el almacén de certificados de confianza (truststore). Además, se deberán mostrar los atributos del certificado temporal, para que el cliente decida si confía en el proveedor de identidad OpenID que proporcionó los atributos con que se generó el certificado temporal.
2. El segundo paso de la verificación consiste en comprobar si los recursos dentro del bundle siguen el orden descrito en las especificaciones: archivo de manifiesto, archivo de hash (Signature File), archivo de firma (Block Signature File) y resto de archivos del bundle
3. El tercer y último paso será verificar la coherencia de los archivos de metadatos [8]: el archivo de firma debe contener una firma válida del archivo de hash, el archivo de hash debe contener un valor hash válido para el archivo de manifiesto y el archivo de manifiesto debe contener el nombre y valor hash de todos y cada uno de los recursos del bundle

En nuestra implementación hemos decidido que la verificación de la firma se haga en el mismo repositorio, de manera que se incorpora el resultado de dicha verificación en la vista que se muestra al usuario de forma que pueda decidir que en bundles deposita su confianza de forma que no supongan una sobrecarga para el dispositivo que va a ejecutar los bundles. Obviamente, esto implica que deba haber una relación de confianza con el servlet del repositorio que se encarga de esta función de verificación de firma si bien en nuestra esquema siempre suponemos que el repositorio es confiable. En cualquier caso, el usuario podrá descargar la firma junto con el bundle y verificarla localmente para un mayor nivel de seguridad.

Por tanto podemos describir el proceso de la autenticidad de los bundles desde el punto de vista de la siguiente manera:

- El cliente se conecta al servicio que muestra el listado de bundles disponibles en el repositorio
- Por cada bundle, el repositorio deberá mostrar:
  - Los atributos propios del bundle, como pueden ser el nombre, el tamaño o la fecha de creación
  - La identidad OpenID del desarrollador/firmante del bundle
  - El proveedor de identidad OpenID (ya que el cliente deberá decidir si es de confianza o no)
  - La CA que certifica dichas identidades,

- Algún tipo de mensaje de error en caso de que se produzca algún tipo de error durante el proceso de verificación.

- Viendo esta información, será misión del cliente decidir si se descarga el bundle y lo ejecuta o no.[3]

V. CONCLUSIONES Y TRABAJO FUTURO

En la actualidad es una máxima que la distribución de software debe de tener en cuenta durante todo su ciclo la seguridad. Entre otros aspectos se deben proporcionar mecanismos para evitar ataques de sustitución de software (o componentes) con el objetivo crítico de impedir la propagación de malware (y, por supuesto, todas las consecuencias nefastas que éste lleva asociado).

La plataforma OSGi tiene en la distribución de componentes software, así como en la actualización, instalación y desinstalación de manera dinámica, autónoma y en tiempo real, su mejor exponente. Nos sirve pues este escenario para desarrollar una plataforma de distribución de bundles (componentes software en OSGi) segura.

Sin embargo, el diseño de la plataforma es generalizable a cualquier escenario de distribución de ficheros (aplicaciones, componentes software, imágenes, etc.) donde el la identidad del propietario sea un factor crítico. La ventaja de nuestro esquema sobre los sistemas tradicionales de firma de código es su flexibilidad y facilidad de uso ya que solo requiere que el desarrollador se autentique frente al repositorio usando un proveedor OpenID favorito. Si bien no se elimina la necesidad de confiar en el repositorio, tal y como ocurre con la distribución de paquetes en sistemas tipo Debian, se añade un grado adicional de seguridad al certificarse la identidad del propietario mediante la firma del bundle con el certificado temporal creado a partir dos atributos proporcionados por el proveedor de identidad.

En la actualidad estamos trabajando en una migración de esta plataforma hacia otros escenarios, y considerando las implicaciones, sobre los terminales en particular y la plataforma diseñada en general, de aumentar los niveles de seguridad de este diseño (por ejemplo mediante el uso de CAs externas para la generación de certificados).

Otro aspecto que estamos considerando es la definición de extensiones para los certificados X.509v3 generados de forma que la cookie de autenticación OpenID o parte de ella se pueda incluir en el certificado temporal de firma.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado a través de los siguientes proyectos: OSAmI (TSI-020400-2009-92) y SPRINT (TIN2009-09237).

REFERENCIAS

[1] Panda Security, Informe Trimestral PandaLabs (Enero-Marzo 2010)
[2] Symbian Developer Certificates (http://wiki.forum.nokia.com/index.php/Developer_certificate)
[3] Secure APT en Debian (http://wiki.debian.org/SecureApt)
[4] OSGi Alliance (http://www.osgi.org)
[5] Eclipse (http://www.eclipse.org)
[6] Pierre Parrend, Protecting code archives with digital signatures, Proyecto OWASP, Enero 2008
[7] OpenID, OpenID Authentication 2.0 – Final, Specification, December, 2007.
[8] How a provider can do self-integrity checking, How to implement a provider for the Java Cryptography Extensions, Java Tutorials, SDK 1.5.
[9] Raffi Krikorian, Programatically Signing JAR Files, O'Reilly on Java, Abril 2001,
[10] JAR Signing and Verification Tool, Java Tutorials (http://java.sun.com/j2se/1.3/docs/tooldocs/win32/jarsigner.html)
[11] Key and Certificate Management Tool, Java Tutorials (http://java.sun.com/j2se/1.3/docs/tooldocs/win32/keytool.html)

[3] El usuario podrá verificar de forma local la firma del bundle para aumentar el nivel de seguridad.

Infraestructura para el Mantenimiento y Evolución de Seguridad y Dependabilidad en Escenarios de Computación Dinámica

Antonio Maña, Rajesh Harjani, José F. Ruiz, Antonio Muñoz
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga
Email: {amg, rajesh, joseruiz, amuñoz}@lcc.uma.es

***Abstract*—Este artículo describe la infraestructura SERENITY [1] para el mantenimiento y evolución de soluciones de seguridad y dependabilidad (S&D) obtenidas dinámicamente. También se presentan las características principales de la infraestructura junto con los diferentes mecanismos que lo conforman y, finalmente, se describe un escenario para ilustrar la aplicación de la infraestructura presentada.**

I. Introducción

La aparición de sistemas altamente distribuidos que operan en ambientes heterogéneos y frecuentemente en condiciones de cambios de contexto (por ejemplo sistemas móviles, sistemas basados en servicios, inteligencia ambiental, etc..), plantean retos significativos para los sistemas de seguridad y dependabilidad (S&D). Esto hace necesario desarrollar mecanismos que soporten la configuración dinámica, el desarrollo, la monitorización y la evolución de S&D Solutions. Con estos ajustes, el mantenimiento dinámico y fluido se convierte en un elemento central para asegurar la seguridad y dependabilidad de los sistemas, manteniendo las soluciones actualizadas y asegurando que son usadas correctamente. El proyecto SERENITY ha desarrollado mecanismos que soportan el desarrollo dinámico, la configuración y monitorización de S&D Solutions que cumplen las propiedades de seguridad genéricas y de dependabilidad y se describen usando modelos llamados S&D Patterns. El modelo SERENITY se centra en la provisión y supervisión de S&D Solutions en tiempo de ejecución, permitiendo la detección de problemas en instancias individuales de S&D Solutions y la reconfiguración automática de las aplicaciones usando estas soluciones. Para aumentar la confianza en las S&D Solutions es necesario analizar su comportamiento cuando se usa en diferentes contextos. Un análisis dinámico global puede detectar situaciones que no son posibles de detectar con un análisis dinámico local o estático, tales como problemas en las implementaciones, en los modelos que los describen o incluso problemas causados por la interacción de soluciones diferentes. Los resultados de este análisis proveen una base para la toma de acciones que puede soportar el mantenimiento y evolución de S&D Solutions en respuesta para identificar problemas. Este artículo describe la extensión del marco de trabajo SERENITY que soluciona este propósito, la Infraestructura de Evolución SERENITY. El objetivo de la Infraestructura de Evolución SERENITY es analizar problemas detectados en las operaciones de S&D Solutions a un nivel global para soportar la reacción automática, evolución y mantenimiento de aplicaciones operando en sistemas altamente dinámicos y heterogéneos. Para conseguir este objetivo, debemos tener en cuenta que por eficiencia y limitaciones de ancho de banda no es posible reenviar todos los eventos recibidos desde las diferentes soluciones. De hecho, las razones de seguridad, de dependabilidad y de privacidad hacen inaceptable reenviar todos los eventos a monitores remotos. Por estas razones, hemos desarrollado una infraestructura de tres capas. La primera capa de la infraestructura usa información de violaciones de reglas generada por el marco de monitorización de SERENITY [1]. Una segunda capa, con objetivo de controlar los problemas relacionados con la interacción de diferentes soluciones en una plataforma dada y dirigida por un componente llamado Agente Transparente. Por último, una tercera capa controla la operación de soluciones específicas que corren en diferentes plataformas, basados en Metamonitores específicos para cada solución. Empezando por las reglas de monitorización locales, la infraestructura está diseñada para evaluar la integridad de las S&D Solutions y para dar soporte al mantenimiento del sistema, permitiendo a los administradores del sistema y los desarrolladores de soluciones mejorar estas soluciones basándose en el comportamiento observado. Como se mencionó, para realizar este objetivo la infraestructura recoge, agrega y analiza información sobre las violaciones de diferentes propiedades a través de distintas instancias del marco SERENITY y las aplicaciones que ofrece. El resto de este artículo está organizado de la siguiente manera. La Sección 2 presenta trabajos relacionados y un transfondo para el modelo SERENITY. La Sección 3 presenta la infraestructura de mantenimiento y evolución SERENITY. En la Sección 4 se describe un escenario de aplicación. Por último, la Sección 5 presenta las conclusiones y trabajo futuro.

II. Transfondo y trabajo relacionado

La infraestructura que presentamos ha sido desarrollado en el último año del proyecto SERENITY. Debido a la novedad

[1]Trabajo realizado parcialmente en los proyectos SERENITY (FP6-IST-027587) y DESEOS (TIC-4257)

del modelo de sistema seguro y dependable definido en el proyecto, no hay otra infraestructura que ofrezca las mismas características que ofrecemos nosotros en este artículo. Por esta razón, no se encuentran muchos trabajos relacionados en esta sección. De hecho, se ofrece algunos enfoques parcialmente relacionados. Además, se ofrece una introducción al modelo SERENITY de sistemas de computación seguros y dependables.

A. Trabajos relacionados

El mantenimiento de software es un conjunto de actividades realizado en un sistema software para su uso operacional. Las encuestas han mostrado que, para muchos proyectos, el mantenimiento de software consume la mayoría del coste del ciclo de vida, y hay indicaciones de que los costes de mantenimiento aumentan proporcionalmente. La detección de violaciones de propiedades esperadas son el punto de inicio del mantenimiento de actividades en el desarrollo software [2]. El trabajo en las arquitecturas de mantenimiento es difícil y obsoleto [3]. Las tecnologías de mantenimiento [4], [5] tienden a ignorar la primera fase de la actividad de mantenimiento, la detección de errores. Además, como en cualquier desarrollo, la confianza en las S&D Solutions que están disponibles y pueden ser usadas a través del marco de trabajo SERENITY es un prerrequisito fundamental. A pesar del reconocimiento de la importancia y necesidad en la confianza en interacciones humanas e intercambios y, como consecuencia, el reciente incremento del volumen de la literatura en este tema (por ejemplo [6], [7], [8], [9]), la confianza está actualmente pobremente evaluada para los propósitos de desarrollo dinámico de software. Además, ninguna de estas hebras de trabajo soluciona correctamente algunos aspectos importantes de la confianza en el servicio software, como en el caso de las S&D Solutions de SERENITY, basándose no solo en opiniones subjetivas si no también en información adquirida dinámicamente sobre el comportamiento y calidad de los servicios software en diversos contextos de desarrollo. Además, cada valoración de la confianza debería estar acompañada por una evaluación de su precisión y riesgo [10].

B. El modelo SERENITY para sistemas seguros y dependables

El objetivo de esta sección es facilitar el entendimiento del modelo subyacente que usamos como base en nuestro trabajo. Sin embargo, debido al tamaño y complejidad, está fuera del objetivo de este artículo dar una completa visión de SERENITY. El lector interesado puede consultar la referencia [1] para una descripción comprensiva del sistema completo. El proyecto SERENITY ofrece un marco de trabajo para el tratamiento automático de las cuestiones de Seguridad y Dependabilidad (S&D) en los escenarios de Inteligencia Ambiental (AmI) concentrándose en dos puntos: (i) capturar la maestría específica de los ingenieros de seguridad de tal forma que permita su procesamiento automático; y (ii) ofrecer medios para realizar selección, monitorización y reemplazo de mecanismos de seguridad y dependabilidad en tiempo de ejecución. Estos dos puntos han sido desarrollados por medio de un conjunto de componentes, descritos a continuación. En SERENITY, el pilar fundamental para construir soluciones seguras y dependables es el concepto de S&D Pattern (Patrón S&D). Existen cinco artefactos para representar de una forma lógica S&D Solutions en el proyecto SERENITY: S&D Classes, S&D Patterns, Esquemas de Integración (Integration Schemes), S&D Implementations y Componentes Ejecutables (Executable Components). Estos artefactos representan S&D Solutions usando descripciónes semánticas a diferentes niveles de abstracción. La razón principal para usar diferentes artefactos, cada uno para un distinto nivel de abstracción, es porque de esta forma es posible cubrir el ciclo de vida completo de las aplicaciones de seguridad, especialmente en las fases de desarrollo y de ejecución. Los S&D Patterns son descripciones precisas de S&D Solutions abstractas. Estas descripciones contienen toda la información necesaria para la selección, instanciación, adaptación y aplicación dinámica de la solucion representada en el S&D Pattern. Están compuestos de diferentes elementos los cuales describen las funcionalidades de los patrones de seguridad y como usarlos. Desde el punto de vista de su implementación, los elementos mas interesantes de la estructura de los S&D Pattern son: (i) la interfaz del patrón, la cual describe las funcionalidades del S&D Pattern y como usarlos; (ii) las S&D Classes a las cuales pertenece el S&D Pattern (ver abajo); y (iii) el ClassAdaptor. Los S&D Patterns incluyen descripciones de como adaptar la interfaz del S&D Pattern al interfaz del S&D Class. Los S&D Patterns representan soluciones monoliticas de S&D Solutions, aunque existe un tipo especial de S&D Solution llamado Integration Scheme (Esquema de Integración), el cual consiste en un S&D Solution que está al mismo nivel que los S&D Patterns. Representan S&D Solutions que son construidas para ser combinadas con otros S&D Patterns. Cuando se está desarrollando una aplicación, los Integration Schemes se usan de forma similar a como lo hacen los S&D Patterns. Sin embargo, difieren en su proceso de desarrollo. A lo largo de este documento se referirá tanto a los S&D Patterns como a los Integration Schemes como S&D Patterns indistintamente. Las S&D Classes representan abstracciones de un conjunto de S&D Patterns caracterizados por proveer las mismas S&D Properties y poseer una interfaz común. Este es uno de los artefactos mas interesantes usados en tiempo de desarrollo por los desarrolladores de sistemas. El propósito principal de este artefacto es facilitar la sustitución dinámica de las S&D Solutions en tiempo de ejecución a la vez que facilita el proceso de desarrollo. Las S&D Implementations representan los componentes que implementan las S&D Solutions. Estos artefactos no son implementaciones reales si no que representan una descripción de la implementación. Estos componentes son accesibles a las aplicaciones a través del SRF. Un S&D Implementation describe una implementación de un S&D Pattern. Un S&D Pattern puede tener mas de una S&D Implementation. Finalmente, los Executable Components son las implementaciones de las S&D Implementations. Estos elementos no son usados en tiempo de desarrollo pero son la implementación real de la S&D Solution. Un Executable

Component funciona como una S&D Solution ejecutable que provee sus servicios de S&D a las aplicaciones. En tiempo de ejecución el SRF es capaz de activar, desactivar y adaptar los Executable Components dependiendo de los requisitos de S&D de las aplicaciones. Los S&D Classes, S&D Patterns y S&D Implementations son artefactos orientados al tiempo de desarrollo y los Executable Components están especialmente diseñados para ser usados en tiempo de ejecución. Dependiendo del nivel de abstracción del artefacto usado por un desarrollador de aplicaciones, en tiempo de ejecución el SRF es mas flexible cuando selecciona las S&D Solutions. El propósito principal de introducir este esquema es facilitar la sustitución dinámica de las S&D Solutions en tiempo de ejecución a la vez que se facilita el proceso de desarrollo. Estos nuevos conceptos pueden ser útiles de dos formas diferentes: en tiempo de diseño/desarrollo y en tiempo de ejecución. En el primer caso, debemos considerar que las aplicaciones grandes desarrolladas hoy en día se construyen integrando soluciones de distintas fuentes y en diferentes niveles de abstracción. Estas aplicaciones se enfrentan a la existencia de amenazas y errores que pueden requerir mantenimiento. El lector puede encontrar los detalles de como implementar aplicaciones SERENITY y S&D Solutions en [12]. Usando el enfoque SERENITY, este mantenimiento se realiza con un esfuerzo mínimo, incluso automáticamente y sin errores. En el segundo caso (por ejemplo en tiempo de ejecución), los S&D Patterns se usan para ofrecer una adaptación automática de los mecanismos S&D al realizar cambios de contexto. Para conseguir este objetivo, es necesario tener un marco de trabajo que ofrezca el manejo de una librería de patrones y la evolución constante de tales patrones, teniendo en cuenta el contexto en el cual se aplican. El SERENITY Runtime Framework (SRF) es capaz de seleccionar la S&D Solution mas apropiada entre las disponibles, basándose en los requisitos de los usuarios y el contexto actual.

III. Infraestructura de mantenimiento y evolución de SERENITY

La arquitectura propuesta se basa en el Runtime Support Model (modelo de apoyo en tiempo de ejecución) de SERENITY. En este modelo, las aplicaciones cuentan con el SRF [1] para que les proporcione las S&D Solutions que necesiten. Para permitir el mantenimiento y evolución de estas soluciones, y por tanto de las aplicaciones que las usan, se ha diseñado una arquitectura multicapa que proporciona todos los mecanismos necesarios para realizar procesos de verificación dinámica. Intrínsecamente a estos mecanismos se encuentra el concepto de transparencia. Cada S&D Pattern está diseñado para garantizar su correcto funcionamiento por medio de la monitorización en tiempo de ejecución. Para llevar a cabo esto, en la descripción de un patrón se especifican unas reglas de monitorización. En el caso de los patrones de SERENITY estas reglas se expresan usando EC-Assertion [13]. EC-Assertion es un lenguaje de lógica temporal de primer orden basado en el Cálculo de Eventos. Las reglas de monitorización en este lenguaje tiene la forma B => H, donde se declara que si B es verdadero, H debe serlo también. Ambos B (Cuerpo) y H (Cabeza) son definidos como conjunciones de predicados en el Cálculo de Eventos. Los predicados que se usan en las reglas de monitorización expresan la ocurrencia de un evento (predicado Happens), el comienzo o fin de una condición por medio de la ocurrencia de un evento (Predicados de Inicio y Fin respectivamente), o la validez de una condición (predicado HoldsAt). Los predicados se asocian a variables de tiempo que indican el momento en el que el predicado es Verdadero. Los eventos en el EC-Assertion puede que representen invocaciones de operación y respuestas, o señales y mensajes generados durante el funcionamiento de un sistema, y están ligadas a una variable de tiempo. La descripción detallada del EC-Assertion no forma parte del objetivo de este artículo, y puede ser encontrada en [13]. Un ejemplo de una regla de monitorización expresada en EC-Assertion es:

```
Happens(e(_id1,
          _sender,
          _receiver,
          REQ,
          decrypt(_x,_y),
          _receiver),
        t1,
        R(t1,t1))

⇒

Happens(e(_id2,
          _receiver,
          _sender,
          RES,
          decrypt(_x,_y),
          _receiver),
        t2,
        R(t1,t1+1000))
```

Esta regla expresa una propiedad de disponibilidad obligatoria para una operación de descifrado decrypt (_x, _y) que descifra una cadena de entrada _x y genera otra de salida _y. Según la regla, después de la invocación de una operación decrypt en el componente de descifrado _receiver en algún momento t1, debe haber una respuesta al invocador de la operación (_sender) en algún momento t2 no más tarde de 1000 milisegundos tras t1. La restricción temporal para la respuesta a la invocación se indica en el rango de tiempo de la variable t2 del evento de respuesta que, como se indica en la regla es R(t1,t1+1000) (por ejemplo, (t1,t1+1000]). La reglas de monitorización en los S&D Patterns deben estar diseñadas para proporcionar la información requerida por la aplicación, la cual usa el patrón para evaluar el correcto funcionamiento del patrón y de los Componentes Ejecutables que lo implementan. En un sistema en funcionamiento, los bloques básicos de construcción son los Componentes Ejecutables (ECs), los cuáles son implementaciones de los S&D Patterns. Los Componentes Ejecutables deben incluir Capturadores de Eventos para informar a sus clientes sobre su funcionamiento interno. Por lo tanto, es de obligado cumplimiento para todas las implementaciones de un S&D Pattern, incluir código para capturar los eventos utilizados en las reglas de monitorización del patrón y notificar estos eventos a la aplicación a través del SRF.

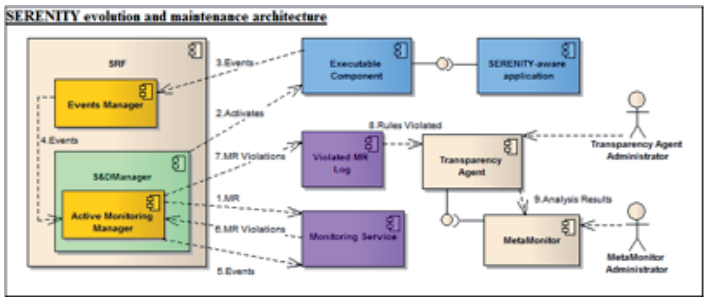


Fig. 1. Arquitectura de evaluación y mantenimiento de SERENITY.

Cuando el SRF recibe un evento de un EC, este evento es gestionado por el mecanismo de monitorización de la plataforma. En este caso, el evento es reenviado al servicio de monitorización oportuno, el cual evalúa el estado actual del EC aplicando las reglas de monitorización definidas en el S&D Pattern correspondiente. Si se detecta la violación de alguna regla, se informa sobre ello al SRF, el cual registra la violación y toma las acciones apropiadas. Estas acciones pueden ser: desactivar un patrón, pausarlo, resetearlo, etc. En [14] podemos encontrar una descripción detallada. Para tratar con posibles problemas causados por la interacción entre distintos ECs, existe un segundo mecanismo de monitorización encargado de monitorizar a nivel de una plataforma SERENITY en concreto. El Agente de Transparencia sirve a este propósito realizando un análisis horizontal. En concreto, analiza los datos provenientes de distintos ECs que se están ejecutando en la misma máquina. El Agente de Transparencia recoge información relacionada con las violaciones de las reglas de monitorización, las analiza y envía los resultados al Metamonitor. Los resultados enviados dependen de ciertas reglas, de manera que es el administrador del TA el que decide qué información se envía al Metamonitor y qué información permanece como secreta. Para permitir la evolución de S&D Solutions específicas y detectar problemas tanto de implementaciones incorrectas, como de problemas en el modelado, existen elementos específicos llamados Metamonitores que llevan a cabo un análisis vertical. Estos analizan información que procede de distintas máquinas sobre la misma S&D Solution. Los Metamonitores reciben información de distintos Agentes de Transparencia y realizan un nuevo análisis de estos datos. Por tanto, el Metamonitor posee un visión global de como es el comportamiento de las distintas S&D Solutions en contextos diferentes, y es capaz de deducir las conclusiones apropiadas. La existencia de Metamonitores beneficia no solo a los usuarios de las soluciones sino que también a los desarrolladores de las mismas.

A. *Elementos de Mantenimiento de la Evolución*

El propósito principal de la Infraestructura de Evolución de SERENITY es posibilitar la formulación de cálculos de confianza basados en la información obtenida en tiempo de ejecución sobre el comportamiento de los componentes que implementan los S&D Patterns y S&D Solutions. El Agente de Transparencia (TA) se asocia a una aplicación cliente específica que hace uso de S&D Solutions a través de una instancia particular de la plataforma SERENITY. El objetivo del TA es el de recoger información relacionada con las violaciones de reglas de monitorización en el contexto de la aplicación cliente con la que se encuentra asociada, y analizar dicha información para identificar problemas relacionados con los casos particulares del uso de S&D Solutions. Los resultados del análisis realizado por el Agente de Transparencia son puestas a disposición del administrador local del sistema, el cual posteriormente puede actuar en consecuencia, y realizar cambios a nivel local, normalmente serán cambios en la configuración de la aplicación cliente y la plataforma SERENITY local. Estos resultados también pueden ser enviados al Metamonitor para permitir un análisis de distintas plataformas SERENITY. El Metamonitor (MM) está bajo el control del proveedor de una determinada S&D Solution, y se ejecuta en una infraestructura externa. El MM recibe los resultados de los análisis que han realizado los distintos Agentes de Transparencia de los distintos SRFs, que usan esa S&D Solution particular. El cometido principal del MM es analizar las violaciones de las reglas de seguridad de una determinada S&D Solution (y su correspondiente S&D Pattern) en distintos contextos operacionales, y usar ese análisis para sugerir posibles formas de mejorar dicha solución y/o el patrón que la describe. Ejemplos de este tipo de análisis incluyen comparaciones del número de violaciones de reglas de monitorización de distintas S&D Solutions, que implementan el mismo patrón en contextos operacionales diferentes o la identificación de puntos en común en condiciones de contexto operacionales. El análisis del MM proporciona la base para la formulación de cálculos de confianza de S&D Patterns y S&D Solutions que se ejecutan en distintos contextos operacionales y que pueden dar lugar a tomar decisiones como la modificación en la descripción de un S&D Pattern, la desactivación de determinadas S&D Solutions que están violando las condiciones de ejecución establecidas por el S&D Pattern en distintos contextos, etc.

IV. Escenario de aplicación

Para poner de manifiesto el uso de la infraestructura de evolución de SERENITY consideramos el escenario en el que dos aplicaciones que se ejecutan en dos empresas diferentes necesitan intercambiar información confidencial. Explicar como construir estos elementos está fuera del alcance de este artículo, pero puede ser consultado en [1]. Cada empresa tiene un nodo SERENITY que contiene un SRF, aplicaciones locales, S&D Solutions, Monitores locales y un Agente de Transparencia. Las S&D Solutions son monitorizadas para detectar violaciones en las reglas a través de los eventos detectados y enviados por los Componentes Ejecutables. Los Agentes de Transparencia monitorizan a las S&D Solutions locales y envían sus estadísticas a los Monitores (que son solution-specific) donde las reglas violadas son analizadas. Para construir este escenario se han desarrollado las S&D Solutions necesarias y las aplicaciones que usan las empresas para el intercambio de información entre ellas. Las S&D

Solutions están almacenadas en el SRF local de cada empresa. Las aplicaciones solicitan a estos SRFs S&D Solutions, y obtienen Componentes Ejecutables que implementan dichas soluciones. Los ECs son monitorizados por medio de los eventos que envían al SRF. Estos eventos poseen información sobre el estado del funcionamiento de estos ECs y pueden informar al SRF cuando una regla en violada. El SRF almacena eventos en un log de eventos, de manera que es el Agente de Transparencia local el que accede al mismo. Parte de esta información es reenviada a los Metamonitores. Supongamos que de forma repetida recibimos en el TA violaciones sobre regla x en EC A y regla y en EC B. Si la frecuencia de estas violaciones es estadísticamente significativa, se podría concluir que existe una interacción inusual en estos dos ECs y en consecuencia tomar las medidas apropiadas (como corregir ambos ECs o detectar la incompatibilidad en la descripción de sus S&D Patterns). Ahora imaginemos que el Metamonitor recibe de forma repetida violaciones de un regla por parte del EC A indicando una transición ilegal entre dos estados. Esto indicaría que el EC A no se ajusta al modelo del S&D Pattern al que corresponde. Por otro lado si las violaciones provienen de distintos ECs del mismo S&D Pattern, esto indicaría un error en el modelo del S&D Pattern. Toda esta información puede ser enviada a los desarrolladores de una determinada S&D Solution de manera que les sea útil a la hora de corregir posibles fallos de dicha solución. Una vez la mejora haya sido realizada, el gestor de seguridad de SERENITY puede notificar a los nodos cliente para que eliminen, actualicen o agreguen una nueva solución de seguridad que mejore el rendimiento de la que se está usando.

V. Conclusiones y trabajos futuros

Este artículo ha presentado la infraestructura de SERENITY para el mantenimiento y evolución de componentes de seguridad y dependabilidad, y de las aplicaciones que las usan, en escenarios de computación dinámica. Esta infraestructura añade dos niveles de monitorización más por encima del modelo base de monitorización de SERENITY para proporcionar soporte al mantenimiento y evolución de estas soluciones y aplicaciones. Actualmente, los resultados de los análisis realizados por la Infraestructura de Evolución se ven reflejados en el comportamiento del SRF a través de cambios realizados de forma manual por un administrador encargado de esta tarea. Como trabajo futuro nuestros esfuerzos van dirigidos a la consecución de la reacción automática por parte del SRF acorde a los análisis realizados por la Infraestructura de Evolución. Otra línea interesante de investigación actualmente iniciada es la identificación de cambios que deben ser aplicados a un componente cuando un problema es detectado.

References

[1] G. Spanoudakis, A. Maña and S. Kokolakis, "Security and Dependability for Ambient Intelligence", en *A*dvances in Information Security, Springer, 2009.

[2] Anir Chowdhury and Scott Meyers, "Facilitating Software Maintenance by Automated Detection of Constraint Violations", Tech. Rep. CS-93-37 Brown Univ., 1993.

[3] Ale Sellink and Chris Verhoef, "An Architecture for Automated Software Maintenance", en *P*roceedings of the Seventh International Workshop on Program Comprehension, 1999.

[4] Chris Verhoef, "Towards automated modification of legacy assets", en *A*nnals of Software Engineering, Volume 9, Issue 1-4. Pages: 315 - 336, 2000.

[5] M.G.J. van den Brand, M.P.A. Sellink and C. Verhoef, "Control flow normalization for COBOL/CICS legacy system", en *P*roceedings of the Second Euromicro Conference on Maintenance and Reengineering, 1997.

[6] L. Corritore, et al., "On-line trust: concepts, evolving themes, a model", en *I*nt. J. of Human-Computer Studies, 58(6): 737-758, 2003.

[7] A. Jsang, "Trust and Reputation Systems", en *F*oundations of Security Analysis and Design IV, FOSAD 2006/2007 Tutorial Lectures, Springer LNCS 4677, 2007.

[8] D.H. McKnight and N.L. Chervany, "The Meanings of Trust", en *T*echnical Report MISRC Working Paper Series 96-04, University of Minnesota, 1996.

[9] P. Resnick et al., "Reputation systems", en *C*ommunications of the ACM, 43(12):45-48, 2000.

[10] G. Spanoudakis, "Dynamic Trust Assessment of Software Services", en *P*roc. of 2nd International Workshop on Service Oriented Software Engineering (IW-SOSE '07), 2007.

[11] R. Needham and M. Schroeder, "Using Encryption for Authentication in Large Networks of Computers", en *C*ommunications of the ACM,21, pp.393-399, 1978.

[12] D. Serrano, José F. Ruiz, A. Muñoz, A. Maña, A. Armenteros, B. Gallego-Nicasio, "Development of Applications Based on Security Patterns", en *S*econd International Conference on Dependability, 2009.

[13] Spanoudakis G, Kloukinas C. Mahbub K., "The SERENITY Runtime Monitoring Framework, In Security and Dependability for Ambient Intelligence, In Security and Dependability for Ambient Intelligence", (eds) Spanoudakis, Mana, Kokolakis, Information Security Series, Springer, ISBN-978-0-387-88775-3 pp. 213-238, 2009.

[14] V. Di Giacomo, C. Kloukinas, D. Presenza, C. Riccucci, G. Spanoudakis, T. Tsigritis, "Dynamic Recovery Mechanisms", en *S*ERENITY Deliverable, A4.D6.1, 2008.

Applying Markov Chains to Web Intrusion Detection

Alejandro Perez-Villegas
Instituto de Física Aplicada
Consejo Superior
de Investigaciones Científicas
Serrano 144 - 28006, Madrid, Spain
Email: alejandro.perez@iec.csic.es

Carmen Torrano-Gimenez
Instituto de Física Aplicada
Consejo Superior
de Investigaciones Científicas
Serrano 144 - 28006, Madrid, Spain
Email: carmen.torrano@iec.csic.es

Gonzalo Alvarez
Instituto de Física Aplicada
Consejo Superior
de Investigaciones Científicas
Serrano 144 - 28006, Madrid, Spain
Email: gonzalo@iec.csic.es

***Abstract*—Nowadays, web applications are target of numerous and varied attacks and their protection is indispensable. In this paper, a simple and effective web application firewall is presented. The system follows an anomaly-based approach, in order to detect both known and unknown web attacks. The system decides whether the incoming requests are attacks or not aided by an XML file, which contains a description of the target web application normal behavior. The XML file defines the features and patterns that valid values of the different elements in the HTTP requests should satisfy. Two models has been built to characterize the features: a model based on the argument lengths, and a model based on the character distribution using Markov Chains. All input requests that deviate from the defined normal behavior are considered anomalous, and rejected by the system.**

The system has been tested using a large set of artificially-generated HTTP requests (both normal and anomalous) targeting a deliberately vulnerable web application. The experiments show that if the XML file has enough data to closely characterize the normal behavior of the target web application, a very high detection rate is reached while the false alarm rate remains very low.

I. Introduction

Web applications are becoming increasingly popular in all sorts of environments, ranging from e-commerce applications to banking. Additionally, web applications handle lot of sensible data, and as a consequence they are subject to all sort of attacks, many of which may be devastating [1]. In order to protect against web-specific attacks, the detection is to be moved to the application layer.

An Intrusion Detection System (IDS) analyzes information from a computer or a network to detect malicious actions and behaviors that can compromise the security of a computer system [2]. Traditionally, IDS's have been classified as either signature detection systems (also called negative approach) or anomaly detection systems (positive approach).

The first method looks for signatures of known attacks using pattern matching techniques against a frequently updated database of attack signatures. It is unable to detect new attacks and in order to work properly, databases must be updated frequently. Signature matching usually requires high computational effort.

The second method overcome these problems, although it is prone to more false positives. It looks for anomalous system activity: once normal behavior is well defined, irregular behavior will be tagged as intrusive. A disadvantage is that in rather complex environments, obtaining an up-to-date and feasible picture of what "normal" network traffic should look like proves to be a hard problem.

Web Application Firewalls (WAF) analyzes the HTTP traffic in order to detect malicious behaviors that can compromise the security of web applications [3].

In this paper, an effective anomaly-based Web Application Firewall (WAF) is presented. This system relies on an XML file to describe what a normal web application is. Any irregular behavior is flagged as intrusive. The XML file must be tailored for each target application to be protected.

Regarding related works, [4] presents an overview of different anomaly detection techniques. Additionally, some works focused on the attack detection in web traffic have been presented: in [5] Kruegel et al. use a parameter-oriented URL format and apply several anomaly-detection models to detect possible attacks. In [6] Markov chains are used for the detection and [7] is an anomaly-based system which infers the type of the request parameters and combines different techniques to detect attacks. Also, ModSecurity [8] is a known commercial WAF which follows a signature-based approach.

The rest of the paper is organized as follows. In Sec. II, a system overview is shown, where system architecture, normal behavior modeling, and attack detection are explained. Section III refers to experiments. Traffic generation, the training phase, the test phase and results are also described. Section IV describes system limitations and suggests future work. Finally, in Sec. V, the conclusions of this work are captured.

II. System Overview

A. Architecture

Our anomaly-based detection system analyzes HTTP requests sent by a client browser trying to get access to a web server. The analysis takes place exclusively at the application layer.

In our architecture, the system operates as a proxy located between the client and the web server. Likewise, the system might be embedded as a module within the server. However, the first approach enjoys the advantage of being independent

of the web platform. A diagram of the system's architecture is shown in Fig. 1.

The input of the system consists of a collection of HTTP requests $\{r_1, r_2, \ldots r_n\}$. The output is a single bit a_i for each input request r_i, which indicates whether the request is normal or anomalous. The proxy is able to work in two different modes of operation: as an IDS and as an IPS.

In detection mode, the proxy simply analyzes the incoming packets and tries to find suspicious patterns. If a suspicious request is detected, the proxy launches an alert; otherwise, it just lets the request continue the way to the server. In any case, the request will reach the web server. When operating in detection mode, attacks could succeed, whereas false positives do not limit the system functionality.

In prevention mode, the proxy receives requests from clients and analyzes them. If the request is valid, the proxy routes it to the server, and sends back the server's response to the client. If not, the proxy blocks the request, and sends back a generic denegation access page to the client. Thus, the communication between proxy and server is established only when the request is deemed as valid.

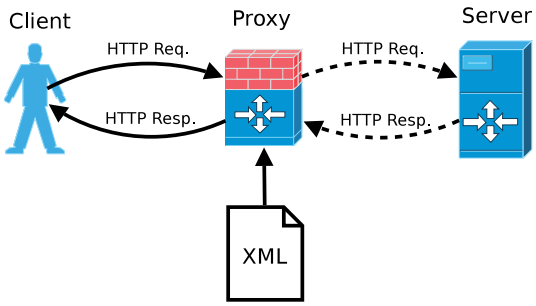


Fig. 1. Web Application Firewall Architecture

B. Normal Behavior Description

Prior to the detection process, the system needs a precise picture of what the normal behavior is in a specific web application. For this purpose, our system relies on an XML file which contains a thorough description of the web application's normal behavior. Once a request is received, the system compares it with the normal behavior model. If the input request does not match the criteria in the XML, it is flagged as an attack and an alert is launched.

The XML file contains rules regarding to the correctness of HTTP methods, HTTP headers, accessed resources, and arguments. This file contains three main nodes:

Methods: The *methods* node simply specifies the list of allowed HTTP methods. Requests using any other method will be rejected.

Headers: The *headers* node specifies a list of the admitted HTTP headers and a description of their values. For each header found in the normal input requests, a *rule* node is included in the XML file. The *rule* node contains information about the header name (attribute *name*) and its description model (subnodes *length* and *markovModel*).

Directories: The *directories* node has a tree-like structure, in close correspondence to the web application's directory structure.

1) Each directory in the web application space is represented in the XML file by a *directory* node, allowing nesting of directories within directories. The attribute *name* defines these nodes.
2) Each file in the web application space is represented by a *file* node within a *directory* node and is defined by its attribute *name*.
3) Input arguments are represented by *argument* nodes within the corresponding *file* node. The attribute *name* is used to define these nodes.
4) In the same way as header values, some properties are used to describe the expected values of the argument. Therefore, every *argument* node has the *length* and the *markovModel* nodes, with their corresponding attributes. Argument values not meeting the length or the Markov model criteria will be rejected.

The adequate construction of the XML file with the suitable properties is crucial for a good detection process. An example of XML configuration file is shown in Fig. 2.

```
<configuration>
<methods>
  <method>GET</method>
  <method>POST</method>
</methods>
<headers>
<rule name="Accept-Language">
<length lowerLimit="0" upperLimit="2"/>
<markovModel A="1.0" Pi="1.0" states="1"/>
</rule>
</headers>

<directories>
  <directory name="shop">
    <file name="index.jsp"/>
    <directory name="public">
      <file name="features.jsp">
        <argument name="id">
<length lowerLimit="0" upperLimit="2"/>
<markovModel A="1.0" Pi="1.0" states="d"/>
</argument>
...

      </file>
    </directory>
  </directory>
</directories>
```

Fig. 2. XML file example

C. Detection Models

In order describe the normal behavior of the arguments and headers, two detection models have been used. The first model is based on the attribute length, and the second one is based on its structure.

1) Attribute Length: The length of an argument, query attribute or header can be used to detect anomalous requests. Many web attacks (code injection, XSS, buffer overflow, etc.) use a considerable amount of input characters. On the other hand, normal requests do not usually contain many bytes.

The system uses a model to learn the normal length distribution for each parameter or header. The model consist of the following. First, we assume that the length of the values for a given input falls into a normal (gaussian) distribution. Second, percentile of the distribution is used to determine a threshold. All lengths that exceed the established threshold are not allowed.

During the training, some statistical properties are collected to describe the normal input length distribution. After collecting the statistical properties, an allowed length interval is calculated, and included in the XML file within the tag `<length>`.

$$[0, \bar{x} + z_{score} * \sigma]$$

- The lower limit is set to 0, since it has no sense to restrict the lower length limit.
- The upper interval is the percentile of the distribution, with a given probability. The percentile p is one of the parameters of the model. In our approach the values $0.90, 0.95, 0.99$ were used. (see Sec. III).

2) Attribute Structure: The structure of the arguments and the headers is also very useful in order to detect anomalous requests. In our approach, a model based on Markov Chains has been used to describe this structure.

A Markov chain is defined by a set of N states $\Gamma = \{S_1, S_2, \ldots, S_N\}$ and by the pair of probability matrices (Π, A). The matrices express the temporal evolution of the system from a statistical point of view. A good text for about Markov chains is [9].

During the training, three features are collected to describe the model: the *states* string, Π and A. These features are included in the XML within the tag `<markovModel>`.

The knowledge about the different *states* reached by the system is obtained from the observation of the training values. These observed states $\Theta = O_i$ can take one or more of these values: l (letters), d (digits) and the rest of printable ASCII characteres (like *,(,),-,', etc.). Letters are grouped in the state l and digits in the state d. From our knowledge about web attacks, we think that by grouping letters and digits, the detection results will be very similar and reducing the number of states, the efficiency of the system increases. Matrices Π and A will only have the states of those characters that indeed appeared in the training set for the corresponding element.

The matrix $\Pi = \pi_i \forall i \in [1, N]$ is a vector that indicates the probability of the i-th state being the first of the temporal sequence of observations.

The matrix A represents the transition probabilities between states. $A = A_{ij} \forall i, j \in [1, N]$: given that the system is in the state i at some time t, probability of it reaching the state j at time $t+1$. The matrix of transition probabilities can be estimated by

$$a_{ij} = \frac{P(q_{t+1} = O_j, q_t = O_i)}{P(q_t = O_i)}$$

In the previous formula it is considered that the state of any trial depends on the state of the directly preceding trial, and only on this state.

The two probabilistic terms in the previous expression can be calculated by counting the number of appearances of the symbol in the corresponding training values. Similarly, Pi can be estimated counting how many times the corresponding symbol is the first one in the observed training value.

Section II-D describes when an input value matches the model. Input values not matching the given model will be rejected.

D. Detection Process

In the detection process, our system follows an approach of the form "deny everything unless explicitly allowed". The detection process consists of several steps, each constituting a different line of defense, in which the proxy checks the different parts of the input request aided by the XML file. If an incoming request fails to pass one of these lines of defense, an attack is assumed. Requests considered as attacks are logged for further inspection. It is important to stress that these requests will never reach the web server when operating in prevention mode. The detection process is composed of the following steps:

1) Method checking. The method must be present in the XML file, otherwise the request is rejected. For example, in the applications in which only GET, POST and HEAD are required to work correctly, the XML file could be configured accordingly, thus rejecting requests that use any other method.
2) Header checking. If the header appears in the XML file, its value must match the length and Markov model criteria. If any header of the request is not valid, it is not accepted, thus preventing attacks embedded in these elements.
3) Resource checking. The system checks whether the requested resource is valid. For this purpose, the XML configuration file contains a complete list of all files that are allowed to be served. If the resource in the request is not present in the list, a web attack is assumed.
4) Argument checking. If the request has any argument, the following aspects are checked:
   a) Allowed arguments. If the request includes arguments not listed in the XML file for the corresponding resource, an illegal manipulation is assumed and thus the request is rejected.
   b) Argument values. An incoming request will be allowed if the actual argument values are identified as normal. For each argument in the request, the actual value is compared with the corresponding Length and Markov models, which describe the

normal behavior of the argument. If a single argument value does not match the models, the whole request is rejected.

For evaluating whether a given observed value is recognized by the previously estimated Markov model, the following approach has been adopted:

Given a Markov chain $\lambda = (A, \Pi)$ and a sequence of observed symbols $O = O_1, O_2, \ldots, O_T$, the sequence is recognized by the Markov chain if the probability of the sequence being generated by the Markov chain ($P[O|\lambda]$) exceeds an stablished threshold.

A useful measure for this purpose is the representation on a logarithmic scale of the Maximum A-posteriori Probability (MAP):

$$LogMAP(O,\lambda) = log(\pi_{O_1}) + \sum_{t=1}^{T-1} log(a_{O_t O_{t+1}}) \quad (1)$$

In this measure, no probability can be zero. A usual technique to solve this is performing a previous *smoothing* of the Markov model. A simple way of smoothing the model is setting those probabilities lower than a given threshold, to a fixed value ε.

While the observations corresponds to the ones expected by the model, the function 'LogMAP' (1) will have no abrupt changes of slope. However, if there is any unexpected symbol, there will be an abrupt change of the slope. To detect these changes and thus the anomalies in the input, the following function, which is an approximation of the derivative of the function 'LogMAP', can be used:

$$D_W(t) = |LogMAP(t) - \frac{1}{W} \sum_{i=1}^{W} LogMAP(t-i)| \quad (2)$$

where W is the window size, and can take values $W = 1, 2, 3, \ldots$. The second term in (2) is the mean of the last W outputs.

(2) supplies an output for each symbol analyzed in the sequence. When the output exceeds a fixed threshold τ, an anomaly is detected.

Different experiments have been performed varying the values of the parameters ε and τ (see Sec.III-E and III-F).

The steps explained before allow the detection of both static attacks, which request resources that do not belong to the application, and also dynamic attaks, which manipulate the arguments of the request.

III. Experiments

A. Case Study: Web Shopping

The WAF has been configured to protect a specific web application consisting of an e-commerce web store, where users can register and buy products using a shopping cart.

B. XML File Generation

As already stated, the XML file describes the normal behavior of the web application.

The names of the different elements of the request (method, headers name, resources and argument names) are obtained from the values in the training set. Therefore, to train the system and to correctly configure this file, only normal and non-malicious traffic to the target web application is required.

Nevertheless, how to obtain big amounts of only normal traffic may not be an easy task, considering that to obtain significative detection results, a large amount of requests are needed. There are several alternatives to obtain normal traffic:

1) Thousands of legitimate users can surf the target web application and generate normal traffic. However, this may not be easy.
2) The application can be published in the Internet. In that case, attacks will be mixed with normal traffic with a very high probability. As explained before, our aim is that only normal requests features are included in the XML file. The better the normal behavior is described, the better the results.

   When working with real traffic, two alternatives can be considered. Either the training phase is performed with the real traffic or the real traffic is filtered before the training phase. The disadvantage of the first option is that real traffic probably contains attacks, thus the system will learn these requests as normal ones and these attacks will not be detected. Filtering the traffic could be very useful, even though it is necessary to be careful as filtering too much could delete normal requests that should be included in the normal description of the web application. Performing a good filtering phase and using a big amount of requests could be a good solution for an adequate training.
3) Traffic can be generated artificially. Although the traffic is not real, we can be sure that only normal traffic is included.

For the e-commerce web application, we considered the third alternative as the most suitable for our purposes hence artificial traffic has been generated for this web application.

C. Artificial Traffic Generation

In our approach, normal and anomalous request databases were generated artificially with the help of dictionaries.

1) Dictionaries: Dictionaries are data text which contain real data to fill the different arguments used in the target application. All the dictionaries used (names, surnames, addresses, etc.) were extracted from real databases.

A set of dictionaries containing only allowed values was used to generate the normal request database, and a different set containing attacks and illegal values was used to generate the anomalous request database.

2) Normal Traffic Generation: Allowed HTTP requests were generated for each page in the web store. Arguments and cookies in the page, if any, were also filled out with values

from the normal dictionaries. The result was stored in database called *NormalDB* containing the normal requests.

3) Anomalous Traffic Generation: Illegal HTTP requests targeting the web store were also generated with the help of anomalous dictionaries. The result was stored in the anomalous request database *AnomalousDB*.

Three types of anomalous requests were considered:

1) **Static attacks** try to request hidden (or non-existent) resources. These requests include obsolete files, session id in URL rewrite, configuration files, default files, etc.
2) **Dynamic attacks** modify valid request arguments: SQL injection, CRLF injection, cross-site scripting, buffer overflows, etc.
3) **Unintentional illegal requests**. These requests should also be rejected even though they do not have malicious intention.

The attacks were generated with the help of the tools Paros [10] and W3AF[11].

D. Training Phase

During the training phase, the system learns the web application normal behavior. The aim is to obtain the XML file from the normal requests collected.

Only normal requests are used to train the system. In our case, the requests used to train the system are part of those stored in the *NormalDB* database.

In the construction of the XML file, different aspects must be taken into account:

- The methods and the resources found in the normal requests are included directly in the XML file as allowed elements.
- Header names are directly included in the XML file. Their values are characterized by extracting the corresponding length and character structure (Markov chain) properties from the requests.
- Likewise, argument names included in the XML and their values are characterized by the corresponding properties (length and character structure) from the requests.

E. Test Phase

During the test phase, the proxy analyzes both normal and anomalous requests. The proxy receives part of the requests from the *NormalDB* database (those not used for the training) and also the requests from the *AnomalousDB*.

The performance of the detector is often measured using Receiver Operating Characteristic (ROC) curves [12]. A ROC curve plots the attack detection rate (true positives, TP) against the false alarm rate (false positives, FP).

$$DetectionRate = \frac{TP}{TP+FN}$$

$$FalseAlarmRate = \frac{FP}{FP+TN}$$

During the test phase, different parameters can be modified to tune the detection models:

- The parameter p corresponds to the percentile used in the length model.
- The parameter ε is used to smooth the Markov model and it is a value assigned to transition probabilities which, in the trained model, are 0 or lower than a threshold. Lower values of ε makes the system more sensible to deviations with respect to the normal behavior.
- The parameter τ is used to decide whether a header/argument value is normal or anomalous. If the value of the parameter is very low, the detection is more prone to false positives. If the parameter is very high, it is not possible to detect attcks with a low level of abnomality.

F. Results

Several series of experiments have been carried out to test the performance of the system, using different combinations of the parameters to tune the detection models. For each series of experiments, the proxy was fed with 10000 normal requests during the training phase. Besides, ten different values of τ were used to obtain the points in each ROC curve: $\tau = \{30, 40, \ldots, 100, 200, 300\}$.

The first series of experiments aims to compare the effect of ε in the Markov detection model. Figure 3 shows the ROC of these experiments, using four different values of ε. The results show that lower values of ε, specifically when $\varepsilon = 10^{-15}$, give better performance.

The second series of experiments aims to compare the effect of p in the lengh detection model. Figure 4 shows the ROC of these experiments, using three different percentiles $p = 0.90, p = 0.95, p = 0.99$. The higher value makes the model more permisive.

In general, the experiments show a very high detection rate. The majority of the attacks given in the input during the test were rejected. As a consecuence of the positive approach, all inputs requesting resources that were not placed in the XML were rejected. This feature is as determinant as the detection models.

On the other hand, about the 20% of the normal requests were tagged as intrusive. While this false alarm rate is too high for real applications, it is reasonably in the experimental scope. The false alarm rate can be reduced by increasing the number of training requests, tuning the detection model parameters and incorporating new detection models to the system.

It is important to notice that when the XML file closely characterizes the web application normal behavior, the different kinds of attacks can be successfully detected and just a few false alarms are raised.

IV. Limitations and Future Work

As shown in the previous section, when the XML file is configured correctly, the system succeeds in detecting any kind of web attack. Thus, the main issue is how to automatically configure the XML description file. In our approach, the XML file is built from a set of allowed requests in the target web application. However, obtaining only normal traffic might not

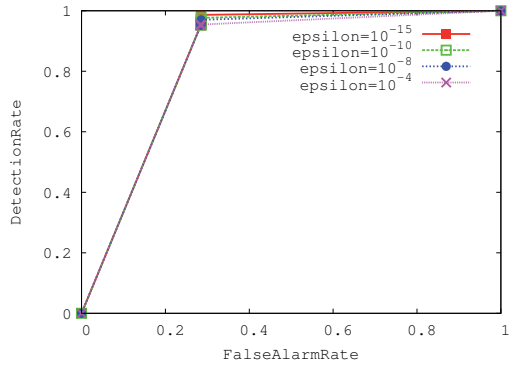


Fig. 3. Comparison of the 4 ROC curves of the WAF for $\varepsilon = 10-15$, $\varepsilon = 10-10$, $\varepsilon = 10-8$, $\varepsilon = 10-4$. In every curve the parameter τ is tuned taking values from 30 to 300. 10000 training requests were used and the percentile is set to 0.99

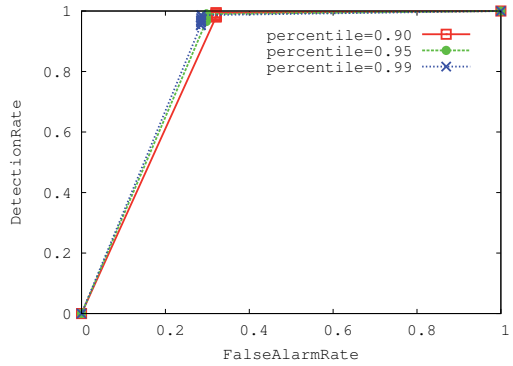


Fig. 4. Comparison of the 3 ROC curves of the WAF for *percentile* = 0.90, *percentile* = 0.95, *percentile* = 0.99. In every curve the parameter τ is tuned taking values from 30 to 300. 10000 training requests were used and ε is set to 10^{-15}

be an easy task, as was discussed in Sec. III-B. Therefore, the main limitation consists in correctly implementing the training phase for any web application.

Other limitations of the system arise when protecting complex web applications. For instance, web sites that create and remove pages dynamically, generate new URLs to access resources, or allow users for updating contents, might difficult the XML file configuration.

As future work, we have several ideas to try to solve the limitations mentioned before and to improve the system. URL patterns can be used in describing sites with dynamic resources in order to solve the protection of complex web applications. We are also working on experiments using real traffic. As real traffic may contain attacks, a filtering phase could be applied to purify the real traffic before the training phase, thus obtaining a more precise description of the web application normal behavior. In addition, more detection models can be included in order to compare their results and conclude which one is more efficient.

V. Conclusions

We presented an efficient web attack detection system or Web Application Firewall (WAF). As the system is based on the anomaly-based methodology it proved to be able to protect web applications from both known and unknown attacks. The system analyzes input requests and decides whether they are anomalous or not. For the decision, the WAF relies on an XML file which specifies web application normal behavior.

The main challenge is how to create an accurate XML file in a fully automated manner for any web application. In our system, we show that inasmuch great amounts of normal traffic are available for the target application, this automatic configuration is possible. The XML file contains a list of the normal HTTP methods, headers, resources and arguments of the web application. The normal values of the headers and the arguments are characterized using Markov chains and length intervals.

Our system has been configured to protect a specific real web application. The experiments for the WAF protecting real web applications show that as long as the XML file correctly defines normality for a given target application, very satisfying results are obtained.

Acknowledgements

We would like to thank the Ministerio de Industria, Turismo y Comercio, project SEGUR@ (CENIT2007-2010), the Ministerio de Ciencia e Innovacion, project CUCO (MTM2008-02194), and the Spanish National Research Council (CSIC), program JAE/I3P.

References

[1] Alvarez G., Petrovic S.: A new taxonomy of Web attacks suitable for efficient encoding. Computers and Security, 22, 5: 453–449 (2003)
[2] Kabiri P., Ghorbani A. A.: Research on Intrusion Detection and Response: A Survey. International Journal of Network Security. 1, 2, 84–102 (2005)
[3] Alvarez G.: Cortafuegos de aplicaciones web (WAF) PC World. 253, 78–84 (2008)
[4] Patcha, A., Park J.: An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks. 51, 12: 3448–3470 (2007)
[5] Kruegel C., Vigna G., Robertson W.: A multi-model approach to the detection of web-based attacks. Computer Networks, 48, 5: 717–738 (2005)
[6] Estévez-Tapiador J., García-Teodoro P., Díaz-Verdejo J.: Measuring normality in HTTP traffic for anomaly-based intrusion detection. Computer Networks, 45, 2: 175–193 (2004)
[7] Bolzoni D., Zambon E.: Sphinx: An anomaly-based web intrusion detection system. Workshop on Intrusion Detection Systems: 14 pages, Utrecht, The Netherlands (2007)
[8] ModSecurity. Open Source signature-based Web Application Firewall. http://www.modsecurity.org (2009)
[9] Feldman R. M., Valdez-Flores C.: Applied Probability and Stochastic Processes. PWS Publishing/Thomson Publishing, USA (1995)
[10] Chinotec Technologies Company: Paros - for web application security assessment. http://www.parosproxy.org/index.shtml (2004)
[11] Andrés Riancho: Web Application Attack and Audit Framework. http://w3af.sourceforge.net (2007)
[12] Provost F., Fawcett T., Kohavi R.: The case against accuracy estimation for comparing induction algorithms. Proceedings of the 15th International Conference on Machine Learning, Morgan Kaufmann, San Mateo, CA, USA (1998)

A Secure Cooperative Sensing Protocol for Cognitive Radio Networks

Carles Garrigues
Estudios de Informática, Multimedia y Telecomunicación
Universitat Oberta de Catalunya
Email: cgarrigueso@uoc.edu

Helena Rifà-Pous
Estudios de Informática, Multimedia y Telecomunicación
Universitat Oberta de Catalunya
Email: hrifa@uoc.edu

***Abstract*—Cognitive radio networks sense spectrum occupancy and manage themselves to operate in unused bands without disturbing licensed users. Spectrum sensing is more accurate if jointly performed by several reliable nodes. Even though cooperative sensing is an active area of research, the secure authentication of local sensing reports remains unsolved, thus empowering false results. This paper presents a distributed protocol based on digital signatures and hash functions, and an analysis of its security features. The system allows determining a final sensing decision from multiple sources in a quick and secure way.**

***Index Terms*—authentication, cognitive radio, cooperative sensing, cryptography**

I. Introduction

Spectrum is an essential resource for the provision of mobile services. In order to control and delimit its use, governmental agencies set up regulatory policies. Unfortunately, such policies have led to a deficiency of spectrum as only few frequency bands are left unlicensed, and these are used for the majority of new emerging wireless applications. Besides, studies conducted by the Spectrum Policy Task Force show that most of the licensed spectrum is largely under-utilized [1].

One promising way to alleviate the spectrum shortage problem is adopting a spectrum sharing paradigm in which frequency bands are used opportunistically. In this scheme, those who own the license to use the spectrum are referred to as primary users, and those who access the spectrum opportunistically are referred to as secondary users. Secondary users must not interfere with primary ones, who always have usage priority.

The enabling technology for opportunistic sharing is cognitive radio (CR) [2]. A CR is a system that senses its electromagnetic environment and can dynamically and autonomously adjust its operating parameters to access the spectrum. CR terminals form self-organizing networks capable to detect vacant spectrum bands that can be used without harmful interference with primary users. Once a vacant band is found, secondary users coordinate themselves in order to share the available spectrum.

Performing reliable spectrum sensing is a difficult task. Wireless channels can suffer fading, thus provoking the hidden node problem in which a secondary user fails to detect a primary transmitter. The most important challenge for a CR is to identify the presence of primary users, and, for this reason, secondary users must be significantly more sensitive in detecting primary transmissions than primary receivers.

In order to reduce the sensitivity requirements of individual CRs, recent studies propose performing distributed spectrum sensing (DSS)[3]. In DSS, multiple secondary users cooperate and share their local sensing results, which are then merged together to reach a final decision. Although the use of cooperation in spectrum sensing has been extensively studied, some security issues still remain unsolved.

The problem of current spectrum sensing protocols is that they do not provide any mechanism to authenticate the observations exchanged by secondary users. This problem is present even in those protocols that intend to deal with malicious users. Secure spectrum sensing protocols assume that sensing reports from secondary users can be effectively authenticated. As a result, malicious users can be detected - their reports repeatedly differ from the final decision- and their contributions discarded. However, a mechanism to authenticate the observations sent by secondary users is still missing.

This paper presents a protocol that enables the secure authentication of sensing information. The protocol is mainly based on the use of hash functions, so that authentication is carried out as quickly as possible. Performing spectrum sensing without significant delay is essential because a lengthy sensing process reduces the time left for transmission. Furthermore, a lengthy sensing process will certainly consume more energy at the CR. Thus, the combination of the proposed protocol with the existing data fusion schemes allows distributed spectrum sensing to be conducted effectively.

II. Background

Cooperative sensing is based on merging the local observations of multiple secondary users. Traditionally, there are two techniques which are used for local spectrum sensing: energy detection or cyclostationary feature detection. Energy detection is based on integrating the energy received over an observation interval. This method is optimal when secondary users do not have sufficient information about the primary user signal. On the other hand, cyclostationary feature detection takes advantage of the fact that signals used in wireless communications are cyclostationary. Thus, their features can

be detected using a spectral correlation function. However this method requires longer observation times.

Since local spectrum sensing results are subject to multi-path and/or shadowing fading, the cooperation among CRs is fundamental to achieve a reliable decision. This cooperation can be implemented in a centralized or distributed manner. In the centralized method, the base station or fusion center (FC) gathers all the information from secondary users and executes the data fusion to reach the final decision. On the other hand, distributed solutions require all secondary users to exchange their local observations, so that the data fusion operation is carried out independently on each secondary device.

Several data fusion schemes have been proposed to merge the sensing data observed by each secondary user. These schemes are based on exchanging of more or less information depending on whether devices perform hard or soft cooperation. When hard cooperation is employed, radios only exchange their final decision: primary user detected or not detected. On the other hand, soft cooperation means that radios exchange their local test statistics with each other. Among the proposed methods, the most typical one is based on applying the "k out of N" rule. This rule determines that the channel is occupied if at least k of the N secondary users have detected the primary signal. As avoiding interference with primary users is a top priority, the most common value of k is 1.

Other methods proposed for merging the sensing data are based on modeling the fusion process as a probabilistic problem. Zarrin and Lim [4] compute the probability of detection by performing the likelihood ratio test (LRT), which is based on the Neyman-Pearson theorem and is used for optimal decision making. Wang et al. [5] apply another probabilistic method where secondary users are classified according to their SNR level and those with higher levels are given more influence on the final decision. Alternatively, Chen et al. [6] propose the use of a Sequential Probability Ratio Test (SPRT). SPRT is a data fusion scheme that supports a variable number of local spectrum sensing results. The protocol assumes that the number of sensing results can be increased and adjusted as necessary, so it guarantees both a bounded false alarm probability and a bounded miss detection probability. The authors also suggest the use of a reputation-based scheme to increase the robustness of the data fusion process.

The introduction of reputation mechanisms in the sensing process has also been considered in some studies. In [7], a two-step protocol for the detection of malicious users that report false sensing data is proposed. In the first step, an outlier detection method is applied to pre-filter those sensing results that are too distant from the rest of the data. In the second step, each user is associated with a trust factor that is based on the past and present sensing data sent by the user. Thus, the trust factor lends more or less weight to a decision depending on the reliability of the corresponding user.

Another important issue to take into account when performing spectrum sensing is preventing primary user emulation attacks. These attacks allow a malicious secondary user to gain priority over others by emulating the signal of a primary user. Solutions to this problem are based on checking whether the estimated location of the transmitter and its signal characteristics match the ones of the licensed primary user [8].

As we have shown, several studies have approached the problem of providing security and reliability to the spectrum sensing process. However, no proposal has been presented so far to authenticate the sensing data provided by a secondary user. Without such authentication, the proposals based on associating a reputation or a probability of detection to each user become useless. The combination of existing protocols with a secure authentication method can undoubtedly improve the performance of spectrum sensing protocols in the presence of faulty radios or malicious users.

III. Protocol

This section presents our protocol for the secure authentication of users' sensing reports. The protocol prevents users from illegitimately claiming false identities and from injecting fake sensing data. Thus, the protocol aims at withstanding the following attacks:

- Altering the final sensing decision. A user could increment her weight in the data fusion process by forging several identities and making a contribution for each of them. With enough forged identities, a user might be able to completely alter the aggregate reading.
- Deceiving the reputation system. By using a different identity each time, a user might report false sensing data repeatedly and avoid earning a bad reputation.
- Obtaining resources unfairly. A user could use many identities to obtain more than her fair share of resources (e.g. bandwidth).

The proposed protocol assumes that the cooperation among CR's is implemented in a centralized manner, which is the most frequently used configuration in the spectrum sensing protocols presented to date. We also assume that the secondary users and the fusion center can use a common control channel.

To perform distributed sensing securely, the cooperative system should identify the users that participate in the sensing process, authenticate their claims, and weigh up their contribution to the final decision based on their reputation or probability of successful detection. Our protocol focuses on the mechanisms required to identify the users and authenticate their sensing results. The final part of the distributed sensing process (i.e. weighing up and merging the contributions) can be implemented using any of the mechanisms that we have mentioned in the previous section. The selection of which data fusion technique to use is out of the scope of this paper.

One of the key goals of the protocol design has been to develop a quick authentication process. We take a public key infrastructure (PKI) approach to identify the peers of the network through digital signatures. Even though this process is costly, it has to be executed only once, in the setup phase. Then, we make use of efficient Hash Message Authentication Code (HMAC) functions to protect users' sensing reports from forgery and manipulation.

HMAC functions provide message authenticity and integrity by calculating a hash of two inputs: the target message and a secret key. In our protocol, we use hash chains to produce one time secret keys. Hash Chains, first proposed by Lamport [9], are versatile low-cost constructions that are used extensively in various cryptographic systems.

The proposed protocol is divided in two phases. The first phase is the identification of users, and the second one is the collection of sensing results. In the following sections, we will describe each one of these phases in detail.

A. Phase 1: user authentication

In the first phase, the user contacts the fusion center (which can be, for instance, the base station) and asks permission to join the cognitive radio network. Besides, she commits to a hash chain by attaching the top value of the chain in the request. This process requires mutual authentication using digital signatures. At this point, the fusion center decides whether or not to accept the user into the network. The following are the detailed steps carried out during this phase.

1) User U chooses a random number w_N and prepares a hash chain of length N, where N is chosen by the fusion center and it is shared by all network members.
   Hash chains are composed of a sequence of values that can only be computed in one-way. A hash chain of length N is constructed by applying a one-way hash function H(.) recursively to an initial seed value w_N: $w_{N-1} = H(w_N)$, $w_{N-2} = H(w_{N-1})$, $\cdots$, $w_0 = H^N(w_N)$. In general, $w_i = H(w_{i+1}) = H^{N-i}(w_N)$.
2) U sends the top value of her chain (w_0) to the fusion center FC in a digitally signed message. The signature is computed using U's private key pvk_U. She also includes information about her identity Id_U (i.e. the unique identifier of her public key certificate).

$$JoinReq = \{w_0, Id_U, Sign_{pvk_U}(w_0, Id_U)\}$$

3) FC verifies the signature received from U using U's public key pbk_U. If the signature is correct, FC decides whether or not to accept U into the network. This decision will be based, for example, on the reputation earned by U in previous processes. The implementation of these mechanisms is out of the scope of this paper.

B. Phase 2: collection of local sensing results

In the second phase, the fusion center requests each user to sense a certain set of frequency bands. Users conduct spectrum sensing using a mechanism based on the energy perceived, cyclostationary statistics, or any other method. Then, they sign their own local sensing results with a HMAC function and send the sensing data and its signature to the fusion center. The keys used to compute the HMACs are taken from the hash chain, so that the fusion center can verify the identity of the sender. The following are the detailed steps carried out in this phase.

1) At time t, FC broadcasts a signed message with a task list ($TaskList$) that contains the list of channels each user has to sense.

$$SensingReq_t = TaskList_t, Sign_{pvk_B}(TaskList_t, t)$$

where

$$TaskList_t = [(Id_0, ChannelList_0, i_0) \cdots (Id_S, ChannelList_S, i_S)]$$

In the above expression, S is the total number of secondary users, and i_j is the hash chain index that points to the value the user j must use in the following step.

2) Each user U verifies the signature of the sensing request and, if correct, senses the channels listed in $ChannelList_U$. After completing the sensing process, each user sends the results $SensingRes$ to FC. These results can be binary decisions or long test statistics, depending on whether hard or soft cooperation is in use. To allow the authentication of the sensing results, these are sent as follows:

$$SignRes_t = SensingRes_t, HMAC(SensingRes_t, w_i)$$

The key used to construct the HMAC is w_i, where i is the index received from FC in $SensingReq_t$.

3) FC waits for the reply of all secondary users and at time t' (with $t' = t + \Delta t$), it generates a new $TaskList_{t'}$. This new task list can contain empty $ChannelLists$ if there are not more channels to sense.

$$SensingReq_{t'} = TaskList_{t'}, Sign_{pvk_B}(TaskList_{t'})$$

where

$$TaskList_{t'} = [(Id_0, ChannelList_0, \{i+1\}_0) \cdots (Id_S, ChannelList_S, \{i+1\}_S)]$$

4) Each user U verifies the signature of the sensing request and creates a reply that depends on whether the corresponding $ChannelList$ is empty or not. If $ChannelList$ is not empty, then the results are constructed as follows:

$$SignRes_{t'} = SensingRes_{t'}, HMAC(SensingRes_{t'}, w_{i+1}), w_i$$

Otherwise, they just contain the key needed to verify the previous HMAC sent to FC in $SignRes_t$.

$$SignRes_{t'} = w_i$$

As can be seen, the response always includes the key w_i used to create the previous signed results sent in $SignRes_t$.

5) FC waits for the reply of all secondary users and verifies the HMACs from the $SignReq_t$. If more channels need to be sensed or more HMACs need to be verified, a new request is generated. Otherwise, FC starts the fusion of the sensing results.

IV. Discussion

The presented protocol provides a way to authenticate sensing reports with a minimum overhead. Each user has to generate a digital signature when she accesses the fusion center for the first time. Afterwards, she only has to validate digital signatures (which is very efficient [10]), and compute HMAC using a costless hash function. The HMAC keys can be generated and checked with efficient mechanisms for fast chain traversal [11], [12] and for economic setup and verification [13], [14]: a one-way chain with N elements only requires log(N) storage and log(N) computation to access an element.

From the security point of view, the proposed system is robust against Sybil attacks, in which a user illegitimately claims multiple identities, and the injection of false sensing reports. Sybil attacks are prevented using certificates generated by a trusted central authority. If a user does not own a valid certificate, she is not authorized in the CR network and can not send sensing reports to the fusion center. On the other hand, the injection of false sensing reports is avoided using verifiable HMAC signed reports.

Reporting a verifiable sensing result involves two user transmissions. First, the user sends the sensing data and its corresponding HMAC. Then, she reveals the HMAC key, which is an element of a hash chain. The fusion center verifies the integrity of the sensing message and the authenticity of the key.

Sensing reports are protected against modification attacks since they are signed with an HMAC. Keys used to compute the HMAC are taken from the secret hash chain w of each user. Therefore, only the user who created the hash chain can compute the corresponding HMAC.

Reply attacks are avoided because each HMAC key is used only once. The fusion center indicates in the sensing request which element i it has to be used. Chain elements are asked in ascending order (w_0, w_1, .. w_N) so knowing a user's previous key gives no information about the present one. Moreover, the sensing request is signed so that an attacker can not modify the requested hash index.

Additionally, as the sensing requests and replies are synchronized by i, it is not effective to block the user's reports in order to steal her keys to later generate fake reports.

V. Conclusion

In this paper, we have identified the security vulnerabilities of a cooperative sensing process and its prejudicial effects in CR networks. We have proposed a secure protocol for centralized based systems that uses digital signatures and hash functions.

The protocol enables the fusion center to verify the identity of network members and to ensure the received sensing information is really originated from the claimed source. One of the main features of the proposal is the fact that is computationally efficient and introduces a small bandwidth overhead. As part of our future research, we plan to integrate reputation measures into the scheme.

Acknowledgements

This work is partially supported by the Spanish Ministry of Science and Innovation and the FEDER funds under the grants TSI-020100-2009-374 SAT2, TSI2007-65406-C03-03 E-AEGIS and CONSOLIDER CSD2007-00004 ARES.

References

[1] Federal Communications Commission. Spectrum policy task force report. ET Docket No. 02-135. Technical report, 2002.

[2] J. Mitola III and G.Q. Maguire Jr. Cognitive radio: making software radios more personal. *IEEE personal communications*, 6(4):13–18, 1999.

[3] S.M. Mishra, A. Sahai, and R.W. Brodersen. Cooperative sensing among cognitive radios. In *IEEE International Conference on Communications*, pages 1658–1663. IEEE Computer Society, 2006.

[4] S. Zarrin and T.J. Lim. Belief Propagation on Factor Graphs for Cooperative Spectrum Sensing in Cognitive Radio. In *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 1–9. IEEE Computer Society, 2008.

[5] W. Wang, W. Zou, Z. Zhou, and Y. Ye. Detection Fusion by Hierarchy Rule for Cognitive Radio. In *Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, pages 1–5. IEEE Computer Society, 2008.

[6] Ruiliang Chen, Jung-Min Park, Y.T. Hou, and J.H. Reed. Toward secure distributed spectrum sensing in cognitive radio networks. *Communications Magazine, IEEE*, 46(4):50–55, April 2008.

[7] P. Kaligineedi, M. Khabbazian, and V.K. Bhargava. Secure cooperative sensing techniques for cognitive radio systems. In *IEEE International Conference on Communications (ICC)*, pages 3406–3410, May 2008.

[8] R. Chen, J.M. Park, and J.H. Reed. Defense against primary user emulation attacks in cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, 26(1):25–37, 2008.

[9] Leslie Lamport. Password authentication with insecure communication. *Commun. ACM*, 24(11):770–772, 1981.

[10] Helena Rifà-Pous and Jordi Herrera-Joancomartí. Cryptographic energy costs are assumable in ad hoc networks. *IEICE Transactions on Information and Systems*, E92.D(5):1194–1196, 2009.

[11] Don Coppersmith and Markus Jakobsson. Almost optimal hash sequence traversal. In *Financial Cryptography*, volume 2357 of *LNCS*, pages 102–119, 2002.

[12] Yaron Sella. On the computation-storage trade-offs of hash chain traversal. In *Financial Cryptography*, volume 2742 of *LNCS*, pages 270–285, 2003.

[13] Markus Jakobsson, Frank Thomson Leighton, Silvio Micali, and Michael Szydlo. Fractal merkle tree representation and traversal. In *The Cryptographers' Track at the RSA Conference (CT-RSA)*, volume 2612 of *LNCS*, pages 314–326, 2003.

[14] Marc Fischlin. Fast verification of hash chains. In *The Cryptographers' Track at the RSA Conference (CT-RSA)*, volume 2964 of *LNCS*, pages 339–352, 2004.

Detección Robusta por Grupos de Señales Primarias en Redes de Radio Cognitiva

Mercedes Jiménez Blasco
Internet Interdisciplinary Institute
Universitat Oberta de Catalunya
Email: mjimenezbl@uoc.edu

José Mut Rojas
Internet Interdisciplinary Institute
Universitat Oberta de Catalunya
Email: jmutr@uoc.edu

Helena Rifà-Pous
Internet Interdisciplinary Institute
Universitat Oberta de Catalunya
Email: hrifa@uoc.edu

***Resumen*—La radio cognitiva es una tecnología inalámbrica propuesta para usar eficientemente los recursos del espectro radioeléctrico permitiendo así reducir la carga existente en las bandas de frecuencia de uso libre. Las redes de radio cognitiva son capaces de escanear el espectro y adaptar sus parámetros para operar en las bandas no ocupadas. Para evitar interferir con usuarios con licencia que operan en un determinado canal, la sensibilidad de las redes tiene que ser muy alta. Ello se consigue con métodos de detección cooperativos. Los métodos de detección cooperativa actuales tienen una carencia de robustez ya sea frente a ataques puntuales o continuos. En este artículo presentamos un método de fusión por grupos que tiene presente el comportamiento de los usuarios a corto y largo plazo. Al realizar la fusión de los datos, el método se basa en dar mayor peso a los grupos de usuarios con mayor unanimidad en sus decisiones. Los resultados de las simulaciones prueban que en presencia de atacantes el método de fusión por grupos propuesto consigue una detección superior a otros métodos, cumpliendo los requisitos de sensibilidad mínimos de las redes de radio cognitiva incluso con un 12 % de usuarios reiteradamente maliciosos o un 10 % de atacantes puntuales.**

I. Introducción

Los numerosos servicios de redes inalámbricas disponibles hoy en día han producido un aumento en la demanda de espectro radioeléctrico. Los recursos de espectro son limitados y están controlados por agencias gubernamentales que otorgan licencias para su uso. Sólo una pequeña parte del espectro se puede usar de forma libre, y esta banda está cada día más sobrecargada. Por contra, el uso de otras frecuencias no supera el 15 %. Así pues, como ha manifestado la Federal Communicacitions Commission (FCC) [3] el reparto y uso actual del espectro es ineficiente.

Las redes de radio cognitiva están surgiendo como una tecnología clave para llevar a cabo una gestión más óptima del ancho de banda disponible [1]. Se caracterizan por tener la capacidad de escuchar el uso que se hace del espectro y adaptar consecuentemente sus parámetros de comunicación para aprovechar los huecos libres que existen en muchas de las frecuencias licenciadas. Ello permite a los usuarios compartir el espectro de forma oportunista y evitar colisiones tanto con los servicios licenciados -televisión, telefonía móvil, etc.-, como con otros usuarios sin licencia que quieren aprovechar el ancho de banda libre de la red. Las entidades y usuarios que ofrecen o consumen servicios con licencia son denominados usuarios primarios dado que tienen prioridad en el uso de la red, mientras que los usuarios sin licencia se conocen como usuarios secundarios. Actualmente el estándar de radio cognitiva que se está desarrollando es el IEEE 802.22 que opera sobre las bandas de frecuencia de los servicios de televisión y está dirigido a formar redes de área regional.

El requisito principal de los sistemas de radio cognitiva es evitar interferir a los usuarios primarios. Sin embargo, dicha tarea es complicada debido a la propia naturaleza del medio inalámbrico. Las señales pueden sufrir desvanecimientos profundos debido al efecto multicamino o porque han atravesado un medio con alta atenuación. Este efecto puede ocasionar el problema del terminal oculto en el que un nodo secundario falla en la detección de una señal primaria. Para evitar errores, la sensibilidad de las radios cognitivas debe ser mucho mayor que la de los receptores primarios. Desarrollar sensores que individualmente garanticen los requisitos de sensibilidad que exige una radio cognitiva es muy costoso. Es por ello que las soluciones comúnmente adoptadas pasan por utilizar otra estrategia: la detección cooperativa [4].

Las técnicas de detección cooperativas combinan los resultados de la monitorización espectral que han realizado varios usuarios secundarios individualmente y obtienen una decisión final acerca de la presencia de un usuario primario en la banda de operación. Dado que el efecto multicamino y el ensombrecimiento son factores locales que degradan la detección de sólo algunos nodos de la red, los esquemas detección cooperativa permiten mitigar dichos efectos, obteniendo así un aumento en la probabilidad de detección del usuario primario. Sin embargo, este paradigma conlleva unos riesgos de seguridad, ya que los nodos pueden reportar información falsa que altere la decisión del estado final del espectro.

Aunque en la literatura hay múltiples propuestas sobre métodos de detección cooperativos, pocos de ellos consideran la presencia de usuarios maliciosos en la red que envíen datos erróneos a propósito. Los que lo hacen, requieren generalmente información *a priori* sobre las condiciones del entorno, ya sea el perfil de los nodos del sistema, las características de la señal y el ruido, la frecuencia de ocupación de los canales, etc.

En este artículo, se propone un nuevo método de fusión de detección cooperativa para radios cognitivas que no asume el conocimiento previo del contexto y que es robusto frente a ataques maliciosos. El algoritmo propuesto utiliza las deci-

siones locales de los múltiples nodos y los divide en cuatro grupos según su factor de aciertos en pasadas detecciones, considerando tanto los resultados obtenidos a corto y largo plazo. Los grupos toman una decisión basándose en la detección de la mayoría de sus miembros. Finalmente, las decisiones de los grupos son fusionadas teniendo en cuenta la reputación global del grupo y su unanimidad en la decisión.

El resto del artículo está organizado de la siguiente forma. En la sección II, se describen aspectos generales sobre la detección cooperativa de señales primarias y sobre métodos básicos de fusión cooperativa. El método de fusión de datos por grupos propuesto se explica en la sección III. En la sección IV, los resultados de las simulaciones verifican el funcionamiento del método propuesto comparado con métodos básicos. Finalmente, la sección V presenta las conclusiones del artículo.

II. Técnicas de detección cooperativas

En esta sección presentaremos, en primer lugar, aspectos generales sobre las técnicas de detección cooperativas y, en segundo lugar, describiremos los métodos de fusión cooperativos básicos más usados.

Una red de radio cognitiva está formada por un grupo de usuarios secundarios que escanean periódicamente su entorno radioeléctrico para detectar la presencia de usuarios primarios. Los usuarios secundarios se encuentran bajo diferentes condiciones de atenuación.

La mayoría de técnicas de detección cooperativas utilizan un centro de fusión que recoge los datos enviados por los nodos secundarios acerca de los resultados de su detección local. El centro de fusión ejecuta un determinado método de fusión sobre los datos para obtener la decisión final.

Los métodos de fusión utilizados por el centro de fusión se pueden clasificar en dos tipos: métodos de fusión de datos multi-nivel (soft-combining) y métodos de fusión de datos binarios (hard-combining). Los primeros fusionan información sobre la medida realizada por cada nodo. La fusión proporciona datos muy ajustados pero el volumen de datos que se requiere que los nodos envíen al centro de fusión es muy elevado. Por otro lado, los métodos de fusión de datos binarios realizan la fusión de las decisiones locales sobre si existe o no usuario primario. Cada una de las decisiones locales se envían al centro de fusión en forma binaria. La principal ventaja de estos métodos es que reducen la cantidad de datos enviados.

Los métodos detallados en este artículo emplean datos binarios para realizar la fusión. Antes de presentar los diferentes métodos propuestos para implementar la detección cooperativa, vamos a describir dos parámetros que son importantes a la hora de evaluar el funcionamiento de una determinada técnica de fusión de datos.

El primer parámetro es la probabilidad de detección y se define como la probabilidad de acierto en la detección de un usuario primario. Esta probabilidad indica cómo de bueno es el método evitando las interferencias al usuario primario. Cuando la probabilidad de detección es alta, conseguimos un nivel elevado de protección de la señal primaria. El segundo parámetro es la probabilidad de falsa alarma y representa la probabilidad de detectar un usuario primario cuando en realidad éste no existe. Cuanto menor sea la probabilidad de falsa alarma, el uso de los canales libres será más eficiente.

II-A. Métodos de fusión de datos binarios

En este apartado introducimos las principales técnicas de fusión de datos binarios existentes para detección cooperativa.

Las reglas OR, AND o Mayoría son los métodos de fusión de datos más básicos y se adaptan a cualquier situación [8]. Estas técnicas deciden sobre la ocupación del canal sumando cada una de las decisiones de los N nodos del sistema (u_i) y comparando el resultado con un umbral. En función del valor del umbral de decisión, estaremos hablando de las reglas AND, OR o Mayoría.

La regla OR declara que el usuario primario está presente si al menos uno de los nodos ha detectado al usuario primario:

$$\text{Si} \begin{cases} \sum_{i=1}^{N} u_i \geq 1 & \Rightarrow \text{ señal primaria presente} \\ \text{Otro;} & \Rightarrow \text{ señal primaria absente} \end{cases}$$

En la regla AND el umbral de decisión para declarar que existe usuario primario es el total de nodos N:

$$\text{Si} \begin{cases} \sum_{i=1}^{N} u_i = N & \Rightarrow \text{ señal primaria presente} \\ \text{Otro;} & \Rightarrow \text{ señal primaria absente} \end{cases}$$

En la regla de fusión por Mayoría se declara el canal ocupado cuando como mínimo la mitad de los nodos hayan detectado al usuario primario:

$$\text{Si} \begin{cases} \sum_{i=1}^{N} u_i \geq \frac{1}{2}N & \Rightarrow \text{ señal primaria presente} \\ \text{Otro;} & \Rightarrow \text{ señal primaria absente} \end{cases}$$

Otra posibilidad para fusionar los datos del análisis del espectro se basa en realizar el Likelihood Ratio Test (LRT) y de ese modo obtener una decisión final óptima. Al modelar el proceso de fusión como un problema probabilístico es necesario disponer de información adicional, a parte de conocer las decisiones locales de los nodos. En particular, se deberán conocer: la probabilidad *a priori* de u_i cuando la decisión final es que no hay usuario primario ($P(u_i|H_0)$) y la probabilidad *a priori* de u_i cuando se decide que existe usuario primario ($P(u_i|H_1)$). El cálculo del LRT se realiza según la siguiente expresión:

$$\prod_i \frac{P\left(u_i \mid H_1\right)}{P\left(u_i \mid H_0\right)} \underset{H_0}{\overset{H_1}{\gtrless}} \lambda$$

donde H_0 representa la hipótesis de que el canal está libre y H_1 de que está ocupado. El resultado del LRT se compara con un determinado umbral λ para obtener la decisión final (H_0 o H_1). Este método deberá ser utilizado en entornos más

estáticos y donde se conozcan determinados parámetros del sistema.

La detección colaborativa permite obtener un análisis de las bandas frecuenciales libres más preciso que una única detección local. Sin embargo, el buen funcionamiento de estos métodos puede estar afectado por los siguientes problemas.

En primer lugar, las señales que reciben los nodos secundarios pueden llegar severamente atenuadas o simplemente puede ocurrir que el terminal secundario no funcione correctamente y realice un análisis del espectro erróneo. Estas causas provocan equivocaciones en la decisión del nodo al detectar la señal primaria.

En segundo lugar, puede ocurrir que el sistema contenga usuarios maliciosos. Este tipo de nodos envían información falsa al centro de fusión alterando los resultados de sus medidas del espectro para alterar la decisión final. Este tipo de ataques conduce a equivocaciones cuando se realiza el algoritmo de fusión de datos. Algunos de los efectos que esto puede producir son el error de falsa alarma o fallo de detección. El caso de falsa alarma reduce el rendimiento del sistema. Sin embargo, el caso de fallo de detección tiene consecuencias más serias porque puede provocar interferencias a los usuarios primarios.

Como resultado de estos problemas recientemente se han investigado nuevos métodos de fusión que implementan contramedidas para atenuar los efectos de los ataques de falsificación de datos o los efectos de equipos defectuosos que inconscientemente envían resultados incorrectos.

Lim et al. presentan un método de fusión de datos binarios que utiliza vectores de confianza y reputaciones [5]. El vector de confianza es un índice asignado por el propio nodo según la confianza que él tiene de la precisión del resultado de sus medidas. La reputación representa la precisión de un nodo en su historial de medidas respecto a las decisiones finales.

En primer lugar, un nodo escanea el espectro, toma una decisión sobre la ocupación del canal, y determina un valor de confianza. Luego, el nodo modifica el valor del vector de confianza con un signo positivo si el nodo decidió que el canal estaba ocupado y con signo negativo en el caso contrario.

A continuación, los nodos envían al centro de fusión sus decisiones locales junto con el nuevo valor de confianza. El centro de fusión agrupa todos los resultados utilizando la regla de fusión por mayoría ponderada con pesos. Los pesos representan la reputación de cada nodo, de manera que se asignan pesos mayores a los nodos más fiables. Por consecuencia, las decisiones de estos nodos contribuyen en mayor medida a la decisión final.

La decisión final u se obtiene según la siguiente expresión

$$u = \begin{cases} 1, & \text{si} \quad \sum_i c_i w_i \geq 0 \\ 0, & \text{si} \quad \sum_i c_i w_i < 0 \end{cases}$$

donde c_i es el vector de confianza para un usuario i y w_i el factor de reputación.

Existen otros métodos de fusión binaria y cooperativos que han sido propuestos con el objetivo de reducir el efecto de nodos maliciosos ([6], [2], [9]) pero su ámbito de aplicación no es tan genérico por requerir el conocimiento de cierta información de entorno.

III. Método de fusión en grupos propuesto

Los métodos de fusión de datos binarios propuestos hasta ahora han sido diseñados para entornos inalámbricos muy estáticos y en los que la presencia de atacantes es limitada. En concreto, sólo se asume la presencia de atacantes de tipo SIEMPRE-SI (siempre dicen que la banda del espectro a analizar está ocupada) y de tipo SIEMPRE-NO (siempre dicen que la banda del espectro a analizar está libre). Sin embargo, nodos que normalmente prestan un buen servicio a la comunidad con una detección fiable de los canales espectrales, pueden sesgar su visión del sistema en el momento en el que son ellos mismos los que necesitan un canal de comunicaciones. Un nodo egoísta puede manipular el sistema y decir que un canal está ocupado cuando en realidad no es así, por el simple hecho de poder ocupar él este canal sin tener que compartirlo con los demás nodos de la comunidad. También puede darse el caso que un nodo malicioso informe que un canal está libre cuando está ocupado, por el simple hecho de interferir y provocar una denegación de servicio a los nodos primarios.

En este artículo, proponemos un método de fusión de datos que tiene en cuenta el comportamiento y la reputación de los usuarios a largo plazo, pero que también está preparado para soportar los cambios bruscos del entorno y por consecuencia, de las decisiones incoherentes de los nodos. Para ello clasificamos los nodos en grupos según su comportamiento pasado. Los grupos toman una decisión basándose en la detección de la mayoría de sus miembros. Finalmente, las decisiones de los grupos son fusionadas dando mayor peso a los grupos de usuarios que más han acertado en pasadas detecciones y que presentan mayor unanimidad de voto en la decisión actual.

III-A. Clasificación de los nodos

Definimos la reputación de un nodo como un valor que mide los aciertos a largo término de sus decisiones de detección, esto es, cuando la decisión local del nodo y la global del sistema coinciden. La reputación $r_i \in [0, 1]$ de un nodo i es:

$$r_i = \frac{\sum_{k=1}^{N_i} a_i}{N_i}$$

donde a_i es el número de aciertos del nodo i sobre un total de N_i detecciones.

Por otro lado, definimos la estabilidad de un nodo como un valor que ilustra los cambios contextuales o de conducta de un nodo en un corto instante de tiempo. Calculamos la estabilidad e_i a partir de los aciertos de detección de un nodo i en un corto espacio de tiempo correspondiente a 4 iteraciones de detección:

$$e_i = \frac{\sum_{k=N_i-3}^{N_i} a_i(k)}{4}$$

donde $a_i(k)$ es una función que retorna 0 o 1 cuando el nodo i en el instante de tiempo k falla o acierta la detección del nodo primario, respectivamente.

A partir de la reputación r_i y la estabilidad e_i, un nodo i obtiene un factor de incidencia $w_i \in [0,1]$ en la decisión:

$$w_i = r_i \cdot e_i$$

III-B. Algoritmo de decisión

Los nodos hacen la detección del espectro y envían al centro de fusión la decisión local $d_i = \{-1, 1\}$ para indicar que la banda está libre (-1) o ocupada (1). El algoritmo de fusión propuesto está basado en la regla de fusión por mayoría, pero en lugar de tratar por igual las decisiones de todos los nodos, el sistema las pondera según el factor de incidencia de los nodos y el grado de unanimidad de la decisión.

En primer lugar el centro de fusión ordena los nodos que participan en la detección cooperativa de forma ascendente según el valor de su factor de incidencia. Luego clasifica los nodos en cuatro cuartiles (G_1, G_2, G_3 y G_4). Los valores de corte de los cuartiles (λ_{12}, λ_{23}, y λ_{34}) vienen determinados por el factor de incidencia de los nodos que ocupan las posiciones al $25\,\%$, $50\,\%$ y $75\,\%$ de la longitud de la lista, respectivamente. Así, los nodos son clasificados en grupos según el valor de su factor de incidencia de la siguiente forma:

$$\text{Si} \begin{cases} 0 \leq w_i \leq \lambda_{34}; & \Rightarrow \quad u_i \in G_4 \\ \lambda_{34} < w_i \leq \lambda_{23}; & \Rightarrow \quad u_i \in G_3 \\ \lambda_{23} < w_i \leq \lambda_{12}; & \Rightarrow \quad u_i \in G_2 \\ \lambda_{12} < w_i \leq 1; & \Rightarrow \quad u_i \in G_1 \end{cases}$$

El centro de fusión agrega los datos reportados por los diferentes nodos de la siguiente manera:

$$\begin{aligned} \gamma \quad = \quad & \overline{G_1} + (1 - \left|\overline{G_1}\right|)\overline{G_2} + (1 - \left|\overline{G_1}\right|)(1 - \left|\overline{G_2}\right|)\overline{G_3} + \\ & (1 - \left|\overline{G_1}\right|)(1 - \left|\overline{G_2}\right|)(1 - \left|\overline{G_3}\right|)\overline{G_4} \end{aligned}$$

con $\overline{G_x}$ el promedio de las decisiones locales recibidas por el grupo G_x.

Siendo $\overline{G_x}$ el promedio de un grupo de valores $\{-1, 1\}$, el rango de valores que puede tomar esta variable está entre -1 y 1. $\left|\overline{G_1}\right| = 1$ cuando la decisión de todos los nodos del grupo G_x sea unánime; $\left|\overline{G_1}\right| = 0$ cuando la disparidad de decisiones entre los nodos del grupo G_x sea máxima, es decir, haya la mitad de los nodos que decidan que la banda del espectro sobre la que han hecho la detección está libre, y la otra mitad de los nodos decidan que está ocupada.

El algoritmo de decisión considera principalmente las decisiones de los nodos que están en el grupo G_1 para tomar la decisión final, ya que son los nodos que gozan de una mayor reputación y tienen estabilidad en el sistema. Sin embargo, cuando las decisiones de este grupo son muy dispares (esto es, el promedio de los datos en valor absoluto es bajo), el peso de este grupo baja y las decisiones de los grupos G_2, G_3 y G_4 toman más fuerza. Como se ha hecho con G_1 se analiza la uniformidad de las decisiones de cada grupo G_x y se pondera consecuentemente la incidencia de este grupo en la decisión final.

La decisión global se toma en función del valor γ resultante. Si $\gamma < 0$ se considera que el canal está libre. Sino, el canal está ocupado.

Una vez el centro de fusión ha tomado una decisión sobre la ocupación del canal, la reputación y la estabilidad de los nodos del sistema es actualizada.

IV. SIMULACIONES

En esta sección se ilustran los resultados de las simulaciones realizadas con el esquema propuesto a través de las curvas ROC. Las curvas ROC representan los pares de probabilidad de detección (sensibilidad) frente probabilidad de falsa alarma para diferentes umbrales de decisión.

Las simulaciones se han realizado con 50 usuarios secundarios distribuidos aleatoriamente sobre una área de $500m$x$500m$ considerando un medio con obstáculos y propagación multicamino. Todos los nodos secundarios utilizan un detector de energía para monitorizar el espectro. Por lo tanto, la SNR que recibe cada usuario es diferente y consecuentemente su capacidad de detección. El centro de fusión utiliza en cada caso uno de los métodos de fusión descritos en las secciones II y III.

Se ha analizado la capacidad de detección para diferentes métodos de detección cooperativos de uso general, que no requieren un conocimiento *a priori* sobre el contexto de aplicación.

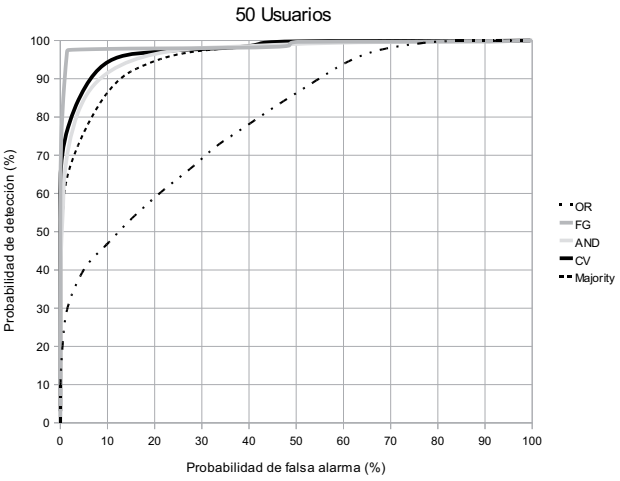


Figura 1. Gráfica ROC. Detección cooperativa con 50 nodos

En la Fig.1 se comparan los diferentes métodos: la regla AND (referida como AND), la regla OR (referida como OR), la regla de fusión por mayoría (referida como Majority), el método con vectores de confianza (referido como CV) y el método de fusión por grupos propuesto (referido como FG). Los resultados muestran que el método propuesto supera al resto de algoritmos de fusión. Para una probabilidad de falsa alarma del $10\,\%$ este esquema consigue una probabilidad de detección mayor al $10\,\%$ que en el esquema de fusión por mayoría.

La curva ROC evidencia que un aumento de la probabilidad de detección va en detrimento de la probabilidad de falsa

alarma y viceversa, lo que implica que la selección del umbral exige un compromiso entre estos dos conceptos. Para conseguir los requisitos mínimos de sensibilidad y probabilidad de falsa alarma estipulados por el estándar IEEE 802.22 los valores deben ser, respectivamente, mayor al 90 % y menor al 10 % [7].

El segundo escenario de simulación que se ha analizado mantiene las mismas características de red que en el caso anterior, pero se han añadido usuarios maliciosos del tipo Siempre-No que atacan reiteradamente. Para evaluar el comportamiento de cada método, se ha fijado un umbral de detección para cada algoritmo que maximice la dupla de la probabilidad de detección y la probabilidad de falsa alarma en un escenario libre de atacantes. A partir de aquí, se ha analizado la probabilidad de detección del sistema a medida que se han ido añadiendo atacantes a la red. La Fig.2 evidencia que la regla AND es el método menos robusto frente a este tipo de ataques, ya que experimenta la mayor disminución. Por otro lado, se observa como el método propuesto cumple el estándar con una probabilidad de detección del 90 % para un porcentaje de atacantes menor al 12 %. Para un porcentaje de atacantes mayor al 20 % no podemos asegurar probabilidad de detección, aunque dependiendo de la configuración de los nodos ésta puede ser positiva. En cuanto a la probabilidad de detección del método de fusión por mayoría y del método con vectores de confianza podemos decir que decrecen suavemente, pero éstos no cumplen el requisito del estándar para una proporción mayor al 2 % de atacantes.

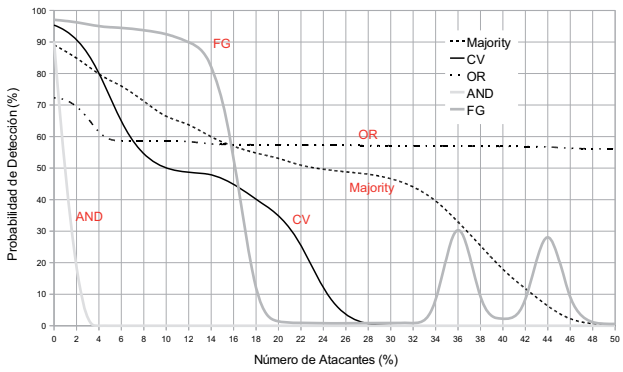


Figura 2. Probabilidad de detección con nodos atacantes

Por último, hemos simulado un escenario con atacantes que actúan por ráfagas. En general los nodos tienen un buen comportamiento en el sistema y por lo tanto gozan de buena reputación, pero puntualmente pueden lanzar un ataque cuando ven que el éxito de éste les puede reportar algún beneficio. Las gráficas que se muestran en Fig.3, Fig.4 y Fig.5, representan la probabilidad de detección del sistema en un periodo de 40 iteraciones. Los nodos tienen un comportamiento correcto durante las 10 primeras iteraciones, pero entre el intervalo de 10 a 30 un determinado número de nodos falsifica sus datos reportando que el canal está libre. Finalmente, después de la iteración número 30, todos los nodos vuelven a enviar al centro de fusión los datos de detección correctos. Las características de la red se mantienen como en los anteriores casos, y siempre hay presencia de un nodo primario en la red.

La Fig.3, la Fig.4 y la Fig.5 representan todas la probabilidad de detección frente al número de iteraciones para un porcentaje del 10 %, del 20 % y del 40 % de atacantes, respectivamente. En las tres gráficas durante el intervalo que se produce la ráfaga de ataques, todos los métodos disminuyen su probabilidad de detección y después de este periodo vuelven a recuperar los valores iniciales.

Comparando las tres gráficas observamos que los métodos de fusión por grupos, con vectores de confianza y por mayoría, empeoran su comportamiento a medida que aumenta la proporción de atacantes. Cuando esta proporción es baja, del 10 % o del 20 %, el método propuesto supera la regla de fusión por mayoría. En particular, para un 10 % de atacantes, el método de fusión por grupos reacciona correctamente manteniéndose en una probabilidad de detección alrededor del 90 % y cumpliendo así las especificaciones del estándar.

Si comparamos este último caso de atacantes puntuales con el caso de atacantes Siempre-No reiterativos (Fig.2), apreciamos que los métodos de fusión sin memoria (como la OR, AND y Mayoría) no presentan comportamientos diferentes entre los dos casos. En cuanto al método con vectores de confianza, tiene probabilidades de detección menores que el método de fusión por mayoría. El método con confianza no está diseñado para tener en cuenta ataques puntuales, el centro de fusión considera correctos los informes que envían los atacantes que habían obtenido buenas reputaciones antes de atacar. El método de fusión por grupos propuesto reacciona correctamente frente a ataques puntuales. Observamos en la Fig.4 como la probabilidad de detección se encuentra alrededor del 70 %. Mientras que para el mismo porcentaje de atacantes reiterativos el método no es capaz de detectar al usuario primario.

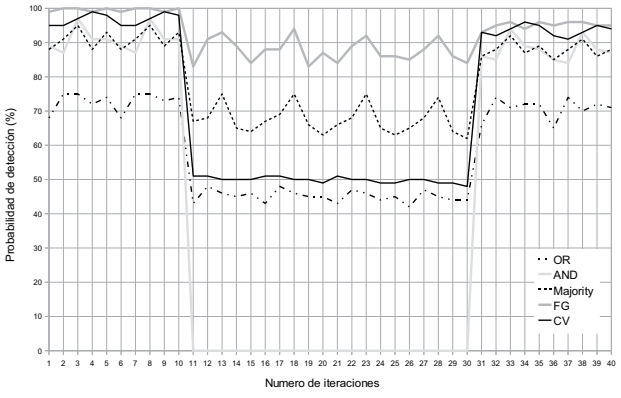


Figura 3. Probabilidad de detección con un 10 % de atacantes puntuales

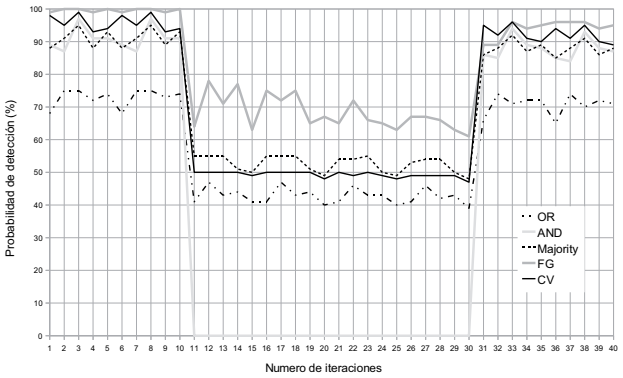


Figura 4. Probabilidad de detección con un 20 % de atacantes puntuales

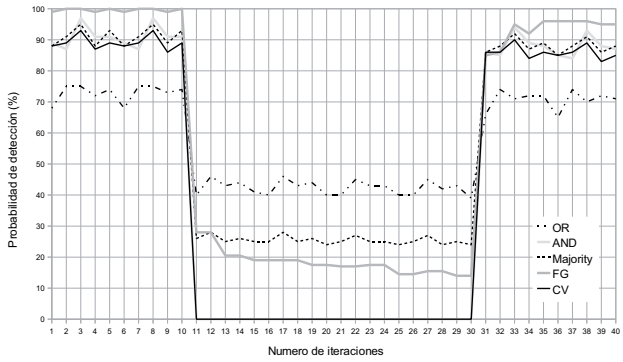


Figura 5. Probabilidad de detección con un 40 % de atacantes puntuales

V. Conclusiones

En este trabajo se ha descrito un esquema de detección colaborativo por grupos con el objetivo de mejorar la sensibilidad y robustez de una red de radio cognitiva. Los resultados de las simulaciones muestran que el algoritmo propuesto ofrece una mejor probabilidad de detección frente a otros algoritmos con unas características de rendimiento similares. En un trabajo futuro se analizará la robustez del esquema en otros contextos y para diferentes volúmenes de nodos.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el Ministerio de Industria, Turismo y Comercio con el proyecto AVANZA TSI-020100-2009-374 SAT2, y por el Ministerio de Ciencia e Innovación y los fondos FEDER con los proyectos TSI2007-65406-C03-03 E-AEGIS y CONSOLIDER CSD2007-00004 ARES.

Referencias

[1] I.F. Akyildiz, W.Y. Lee, M.C. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey. *Computer Networks*, 50(13):2127–2159, 2006.

[2] Ruiliang Chen, Jung-Min Park, and Kaigui Bian. Robust distributed spectrum sensing in cognitive radio networks. In *INFOCOM. The 27th Conference on Computer Communications. IEEE*, pages 1876–1884, April 2008.

[3] S.P.T. Force. Spectrum policy task force report. *Federal Communications Commission ET Docket 02*, 135, 2002.

[4] A. Ghasemi and E.S. Sousa. Opportunistic spectrum access in fading channels through collaborative sensing. *Journal of Communications*, 2(2):71, 2007.

[5] Sunmin Lim, Hoiyoon Jung, and Myung Sun Song. Cooperative spectrum sensing for ieee 802.22 wran system. In *Proceedings of 18th Internatonal Conference on Computer Communications and Networks (ICCCN)*., pages 1–5, Aug. 2009.

[6] Tao Qin, Han Yu, Cyril Leung, Zhiqi Shen, and Chunyan Miao. Towards a trust aware cognitive radio architecture. *SIGMOBILE Mob. Comput. Commun. Rev.*, 13(2):86–95, 2009.

[7] C.R. Stevenson, C. Cordeiro, E. Sofer, and G. Chouinard. Functional requirements for the 802.22 WRAN standard. *IEEE 802.22-05/0007r48*, pages 802–22, Nov. 2006.

[8] E. Visotsky, S. Kuffner, and R. Peterson. On collaborative detection of tv transmissions in support of dynamic spectrum sharing. In *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*., pages 338–345, Nov. 2005.

[9] Wenzhong Wang, Weixia Zou, Zheng Zhou, and Yabin Ye. Detection fusion by hierarchy rule for cognitive radio. In *Cognitive Radio Oriented Wireless Networks and Communications. CrownCom. 3rd International Conference on*, pages 1–5, May 2008.

Uso de rutas cacheadas en el encaminamiento seguro basado en DSR

José Luis Tornos
Centro Politécnico Superior
Universidad de Zaragoza
Email: jltornos@unizar.es

José Luis Salazar
Centro Politécnico Superior
Universidad de Zaragoza
Email: jsalazar@unizar.es

Joan Josep Piles
Centro Politécnico Superior
Universidad de Zaragoza
Email: jpiles@unizar.es

***Resumen*—DSR is a simple and efficient routing protocol in ad hoc networks. This paper is based on a previous variation of the DSR protocol where security is added using aggregate signatures. Our proposal uses an additional Route Discovery Feature of the original protocol, to reduce the number of messages needed around the network to answer a Route Request. We describe how the cached routes of the intermediate nodes can do this work without losing the security level.**

I. INTRODUCCÍON

Una red ad-hoc es aquella en la que no existe ninguna infraestructura de comunicación definida. Normalmente se define sobre dispositivos móviles y a la falta de infraestructura se añade la movilidad de los dispositivos que la conforman, por lo que el sistema es dinámico y admite variaciones.

Debido a la falta de infraestructura predefinida y al dinamismo del sistema, para la comunicación entre nodos sin conexión directa se requiere la cooperación de los nodos intermedios. Existen múltiples protocolos específicos para este tipo de redes ([1], [2], [3], [4], [5], [6]), en los que los métodos de encaminamiento tradicionales no resultan eficientes debido al dinamismo de los dispositivos. Estos protocolos son vulnerables al no tener en cuenta la seguridad de los mismos. Para entornos en los que la seguridad es un requisito importante se han definido protocolos de encaminamiento seguro ([7], [8], [9], [10], [11]), en los que existe un compromiso entre nivel de seguridad ofrecido, ancho de banda, necesidad de procesamiento y potencia necesaria.

Este artículo se fundamenta en el protocolo DSR [12] y profundiza en el planteamiento de un encaminamiento seguro basado en este protocolo [11]. En este esquema se emplean las firmas agregadas como primitiva criptográfica que permiten que M usuarios firmen M mensajes diferentes y las M firmas resultantes sean compactadas en una sola. Gracias a esta característica se consigue reducir el tamaño del campo de firmas, y por tanto, el tamaño del mensaje que se transmitirá. Como contrapartida tendremos que la verificación deberá ser realizada sobre el conjunto de las firmas, no pudiendo verificarse cada una de ellas de modo individual. Es decir, si la firma final es correcta, se validan todos los mensajes que la componen, pero si es incorrecta, no seremos capaces de saber qué mensaje, o mensajes, han sido los que la han invalidado.

El aporte que realiza este artículo, es el uso de las rutas cacheadas para responder a los paquetes de solicitud de ruta, *Route Request (RR)*, opción recogida en el protocolo DSR, manteniendo las características de seguridad preestablecidas. Se desarrolla un método para emplear las rutas cacheadas de los nodos intermedios para ofrecer un encaminamiento seguro que reduciría los mensajes que circulan por la red. También se conseguirá reducir el tiempo necesario para la obtención de una ruta, en el caso de que un nodo intermedio disponga de una ruta cacheada hacia el nodo destino requerido en el paquete *RR*.

En la sección II se presenta brevemente el protocolo DSR y las firmas agregadas. En la sección III desarrollamos nuestra propuesta. Y en la sección IV recogemos las conclusiones y las líneas futuras de trabajo.

II. BACKGROUND

A continuación vamos a recordar los dos elementos fundamentales sobre los que se basa nuestra propuesta: el protocolo de encaminamiento DSR seguro y la primitiva criptográfica de firmas agregadas.

II-A. DSR seguro

Aunque no existe ninguna especificación estándar del protocolo DSR seguro, sí que hay alguna propuesta sobre la mesa basado en DSR ([9], [10], [11]). Empezaremos recordando el DSR básico, para luego explicar la versión segura de [11].

II-A1. DSR: DSR es un protocolo de encaminamiento en el que el nodo origen establece la ruta a seguir en la red hasta alcanzar el nodo destino. Por tanto, en el mensaje de datos enviado están listadas las direcciones de los nodos intermedios que debe atravesar el paquete. Si un nodo tiene que comunicarse con otro para el que no conoce una ruta, envía un paquete *RR*, que será retransmitido por los nodos de la red hasta alcanzar su objetivo. Cuando el paquete *RR* llega al nodo destino, éste responderá con un paquete de respuesta, *Route Reply (REP)*, en el que aparecerán los nodos intermedios que deben procesar los paquetes para comunicar los nodos extremos.

Cada nodo mantendrá una tabla con las rutas conocidas, a las que se añade una marca temporal. Si una ruta no es utilizada dentro del margen temporal será borrada de la tabla de rutas conocidas.

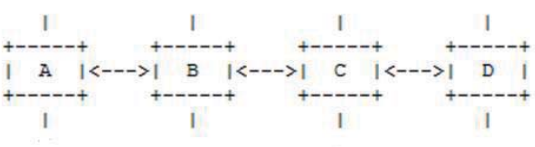


Figura 1: Distribución de nodos en la red ejemplo.

II-A2. Resumen del protocolo DSR seguro: El esquema descrito en [11] para el descubrimiento de rutas de modo seguro, establece que el nodo que quiere obtener una ruta hacia otro nodo, genera un paquete *RR* que firma y propaga en modo multidifusión (broadcast). Los nodos que reciben el mensaje comprueban si son el nodo destino y en caso de no serlo añaden su dirección a la ruta, firman y agregan su firma a la anterior y propagan el mensaje en modo multidifusión.

Cuando el mensaje llega al nodo destino, éste comprueba la firma agregada. Si es correcta, crea un paquete *REP* con la ruta que especificaba el paquete *RR* y lo envía por el camino inverso al que se ha recibido. Cada uno de los nodos intermedios recibe el paquete *REP*, agrega su firma al mensaje y lo envía al siguiente nodo de la lista. Cuando el mensaje llega al nodo que originó el paquete de descubrimiento de ruta, comprueba la validez de la firma agregada y, en caso de ser correcta, añade la ruta recibida a su tabla de rutas.

II-A3. Firmas agregadas: Las firmas agregadas son un concepto criptográfico relacionado con las multifirmas [13]. En el caso de las firmas agregadas, el escenario es un conjunto de usuarios U, cada uno de ellos con su par de claves pública y privada (K_{u+}, K_{u-}). Entonces, dado un subconjunto U'$\subseteq$U, cada usuario u$\in$U', produce una firma σ_u de un mensaje M_u, distinto para cada usuario. Las firmas obtenidas podrán agregarse formando una única firma. Para verificar la firma agregada será necesario tener el acceso a las claves públicas de los usuarios que han firmado cada uno de los mensajes, así como a sus correspondientes mensajes.

Se han desarrollado firmas agregadas en paralelo [14] y secuenciales [15]. Las firmas agregadas en paralelo permiten su verificación sin tener en cuenta el orden en el que se realizó la agregación. Mientras que las firmas agregadas secuenciales deben ser verificadas en el mismo orden en el que se realizó la agregación. Existe otro tipo de esquema de firmas agregadas basadas en identidad [16]. Este esquema elimina la necesidad de certificados pero requiere de una entidad confiable maestra. Otra característica de las firmas agregadas es la longitud de la firma final, que no se mantiene constante para todas las implementaciones. La propuesta elegida es la descrita por Boneh [14], basada en aplicaciones bilineales, y que sí mantiene constante la longitud de la firma agregada e implementa un esquema de firma en paralelo.

III. Empleo de rutas cacheadas en DSR seguro

Partimos del protocolo ya descrito [11] y de sus mecanismos para realizar el descubrimiento de rutas. Al emplear criptografía de clave pública, cada nodo firmará los paquetes, antes de proceder a su retransmisión, con su clave privada. Supondremos la existencia de una autoridad de certificación confiable AC, cuya clave pública es conocida por todos los nodos. La gestión de los certificados y sus revocaciones queda también fuera del ámbito de este trabajo, existiendo diversos esquemas ([17], [18], [19]) que pueden ser utilizados.

Con estas premisas, se va a explicar un método con el que se aprovechan las rutas cacheadas de los nodos intermedios para elaborar paquetes de respuesta de descubrimiento de ruta. Con ello, se consigue una respuesta más rápida y con una necesidad menor de recursos que los requeridos sin esta opción y manteniendo el nivel de seguridad previo.

III-A. Estructura de la tabla de rutas cacheadas

Para poder utilizar las rutas cacheadas para responder a las solicitudes de ruta de otros nodos, deberemos almacenar el paquete con el que se haya verificado esa ruta. También será necesario almacenar las direcciones de los nodos origen y destino que originen el paquete así como el número de identificación de ruta asociado al mismo. La necesidad de almacenar estos valores radica en que son necesarios para elaborar la firma de los paquetes, como se explica en el siguiente apartado.

Además los nodos no añadirán rutas a su tabla solamente cuando ellos son los que originan un paquete de *RR*, sino que cuando reciban paquetes *RR* y *REP*, también serán capaces de actualizar su tabla de rutas, siempre y cuando la verificación de las firmas sea correcta.

Usaremos como ejemplo la distribución de nodos de la Figura 1 y que el nodo A inicia un descubrimiento de ruta hacia el nodo D. Cuando el nodo C reciba un paquete *RR* firmado por los nodos A y B, tendrá un paquete con el que será capaz de validar una ruta hasta el nodo A a través de B. De este modo, si no la conociese de antemano, añadiría a su tabla de rutas el camino para comunicarse con los nodos A y B.

Esto que ocurre con los paquetes de *RR* para los nodos intermedios, también puede usarse con los paquetes *REP*. Por ejemplo, el nodo B recibe el paquete *REP*, que ha enviado el nodo D, y que ha pasado por el nodo C. Recibe el paquete en el que indica que la ruta para llegar del nodo A al nodo D es a través de B, el mismo, y después los nodos C y D. Al estar el paquete firmado por los nodos C y D, será capaz de verificar la validez de esa ruta para esos nodos. Así que será capaz de actualizar su tabla de rutas hacia los nodos C y D gracias al paquete *REP*.

III-B. Datos firmados de cada paquete

Independientemente del tipo de paquete, *RR* o *REP*, se firmarán siempre los siguientes datos y en este orden:

- Número de identificación de descubrimiento de ruta
- IP origen del nodo que ha iniciado el descubrimiento de ruta
- IP destino del nodo objetivo del descubrimiento de ruta
- IP de los nodos intermedios

K_{A-}	Clave privada del nodo A
K_{A+}	Clave pública del nodo A
$\{d\}K_{A-}$	Datos d firmados por el nodo A
$\{d\}K^M{}_{ABC-}$	Datos firmados por los nodos A,B y C y compactados en una firma agregada
$\{d\{d'\{d''\}\}\}K^M{}_{ABC-}$	Datos firmados por los nodos A,B y C y compactados en una firma agregada. El nodo A habrá formado d, el nodo B habrá firmado dd' y así sucesivamente

Tabla I: Lista de abreviaciones

Con este formato, se unifica la estructura de los datos sobre los que se calcula la firma independientemente de si el paquete es *RR* o *REP*.

Dependiendo de si el paquete que se está enviando es un *RR* o un *REP*, tendremos que los nodos intermedios componen una ruta completa entre los nodos origen y destino, para los paquetes *REP*, o una ruta parcial que comunica el nodo origen con el último de los nodos intermedios que componen la lista del paquete, en el caso de los paquetes *RR*.

Un ejemplo del intercambio de mensajes, utilizando la distribución de los nodos de la Figura 1, en el que el nodo A quisiera obtener una ruta hasta el nodo D, sería el siguiente:

A $\rightarrow$ multidifusión: RR $\|$ [N_A, IP_A, IP_D] $K^M{}_{A-}$
B $\rightarrow$ multidifusión: RR $\|$ [N_A, IP_A, IP_D, IP_B] $K^M{}_{AB-}$
C $\rightarrow$ multidifusión: RR $\|$ [N_A, IP_A, IP_D, IP_B, IP_C] $K^M{}_{ABC-}$
D $\rightarrow$ C: REP $\|$ [N_A, IP_D, IP_A] $K^M{}_{D-}$ $\|$ N_A
C $\rightarrow$ B: REP $\|$ [N_A, IP_D, IP_A, IP_C] $K^M{}_{DC-}$ $\|$ N_A
B $\rightarrow$ A: REP $\|$ [N_A, IP_D, IP_A, IP_C, IP_B] $K^M{}_{DCB-}$ $\|$ N_A

Junto al envío de los paquetes *REP*, se hará necesario el envío del valor del identificador de descubrimiento de ruta al que se responde, ya que este valor, que no se envía dentro del paquete *REP*, es necesario para validar la firma.

III-C. Descubrimiento de rutas mediante rutas cacheadas

En el ejemplo descrito en el apartado anterior, si se permitiese el uso de las rutas cacheadas para responder a los paquetes de descubrimiento de ruta y el nodo B tuviese una ruta almacenada para llegar al nodo D, el intercambio de mensajes se habría reducido hasta quedar de la siguiente manera:

A $\rightarrow$ multidifusión: RR $\|$ [N_A, IP_A, IP_D] $K^M{}_{A-}$
B $\rightarrow$ A: REPC $\|$ [*Firma de REPC*] $\|$ [*Datos verif. firma*]

Se aprecia que con esta opción disminuiría el número de mensajes que tienen que ser enviados por la red. La firma que acompaña al paquete *REPC*, Replay cacheado, tiene la misma longitud que el resto de firmas del sistema. Y los datos extra necesarios para la verificación de la firma serán dos direcciones IP y un número de identificación de ruta. Por lo que el tamaño de los datos extra enviados será siempre mucho menor que el necesario para la transmisión de un único paquete DSR por la red.

Para la implementación de esta propuesta se ha diseñado un tipo de paquete cuyo contenido habilite a los nodos la verificación de firmas que asegure la fiabilidad de las rutas.

III-C1. Contenido de REPC: El paquete *REPC* se compone de dos paquetes *REP* dentro del mismo mensaje DSR. Esta opción está recogida dentro del RFC que define el protocolo DSR [12]. El primero servirá para validar la ruta que se ha seguido mediante el paquete de descubrimiento de ruta y el segundo validará la ruta desde el nodo intermedio hasta el nodo final, es decir, la ruta cacheada. Este segundo paquete deberá ser el que el nodo intermedio usó para validar la ruta entre él mismo y el nodo destino.

La variación que habrá que realizar sobre la opción ya recogida en el RFC, será la utilización de un bit, dentro de la zona reservada, que indique al nodo que reciba este tipo de paquetes que la forma de procesarlo será diferente a la recepción de un paquete *REP*. Otra opción sería utilizar un identificador de tipo, que indicase la función del mensaje.

III-C2. Procesado de REPC: Cuando un nodo reciba un paquete REPC sabrá que va a estar compuesto por dos paquetes *REP* y que el primero de ellos deberá procesarlo de manera habitual, sabiendo que la confirmación de ruta será parcial. El segundo paquete *REP* incluirá una ruta que permita alcanzar el nodo destino desde el nodo intermedio que respondió al *RR*.

De manera genérica, la ruta que contiene el segundo paquete *REP* no incluirá exclusivamente los nodos que conecten el nodo intermedio con el nodo destino, sino que formará parte una ruta mayor. Esto implica que la ruta a validar incluya nodos para los que no se ha solicitado información, pero que serán necesarios para realizar la comprobación de la firma del mensaje.

La diferencia existente entre usar una ruta cacheada mediante un paquete *RR* o *REP*, será que en un paquete *REP* se firma la ruta final entre dos nodos y en uno de *RR* se firma una ruta parcial entre los nodos de origen y destino. A la hora de utilizar estos paquetes para añadir una ruta a la tabla de rutas cacheadas de un nodo no existe ningún problema. Pero al utilizar estos paquetes como respuesta en un paquete *REPC*, el nodo que ha solicitado el descubrimiento de ruta deberá ser capaz de diferenciar entre uno y otro, ya que los datos empleados para la firma varían dependiendo de si es un *RR* o un *REP*.

Destino	Saltos	Paquete y firma con el que se validó la ruta	bitC
A	–	RR ‖ [N_A, IP_A, IP_D] $K^M{}_{A-}$	0
C	–	REP ‖ [N_B, IP_D, IP_B, IP_C, IP_D] $K^M{}_{DC-}$	0
D	C	REP ‖ [N_B, IP_D, IP_B, IP_C, IP_D] $K^M{}_{DC-}$	0

Tabla II: Rutas cacheadas del nodo B

Este detalle el nodo destino del *REPC* lo soluciona al inspeccionar el paquete que responde con la ruta cacheada ya que, si al analizar las direcciones IP de los nodos, la última dirección IP de la ruta intermedia, se corresponde con el nodo origen, sabrá que se ha creado a través de un paquete *REP*. Si por el contrario el nodo que se indica como destino no aparece en la lista de nodos intermedios, se tratará de un paquete *RR*.

Una vez interpretado el segundo paquete *REP*, el nodo receptor será el encargado de extraer la información parcial de la ruta que requiere. Para esto deberá localizar el nodo que originó el paquete *REPC* e ir añadiendo a la ruta validada en el primer *REP* los nodos hasta alcanzar el nodo destino.

La posible aparición de bucles en la ruta final se evita empleando los mecanismos ya descritos para ello en el empleo de rutas cacheadas no seguras. Es decir, el nodo que responde a un *RR*, verifica que la ruta que va a proporcionar está libre de bucles. En caso de no disponer de una ruta cacheada con la que no forme bucles, procesará el paquete *RR* como un nodo intermedio normal y lo retransmitirá en modo multidifusión.

Para evitar problemas en los que el tamaño y procesado del paquete *REPC* aumenten en exceso, no se permite a los nodos utilizar rutas aprendidas mediante rutas cacheadas para originar paquetes *REPC*. En la tabla de rutas cacheadas se marcará un bit en aquellas rutas que hayan sido formadas empleando este mecanismo, bitC.

En cuanto a la nueva firma agregada se construye de forma análoga a [11]. Agregamos dos firmas construídas con los mismos parámetros criptográficos que contienen varias firmas agregadas. Dado que estamos trabajando sobre un grupo abeliano, las operaciones cumplen la propiedad asociativa y por lo tanto las firmas también. De esta manera obtenemos una firma de longitud igual a las dos que agregamos. Para verificar dicha firma sólo necesitaremos recopilar las claves públicas de cada uno de los signatarios agregados y los datos extra necesarios para reconstruir los mensajes originales firmados. Con esto la comprobación de la firma se convierte en un simple proceso de verificación de una firma agregada.

Otra de las cuestiones que se podrían plantear sería que una comunicación entre A y B no garantiza que se pueda realizar la comunicación entre B y A. Pero tomamos como supuesto que la comunicación se produce en modo bidireccional.

III-C3. Ejemplo de uso de ruta cacheada: Para explicar la generación del paquete de respuesta *REPC* se va a explicar empleando la distribución de nodos de la Figura 1. Y el caso en el que el nodo A inicia el descubrimiento de ruta hacia el nodo C. Para ello genera y transmite el mensaje formado por el paquete *RR* y la firma del mismo:

$$A \rightarrow \text{multidifusión: RR} \parallel [N_A, IP_A, IP_C]\ K^M{}_{A-}$$

Cuando le llega el paquete al nodo B, este busca en su tabla de rutas cacheadas, Tabla 2, y como tiene una ruta almacenada para llegar al nodo C, inicia la generación del paquete *REPC*.

Primero crea un paquete REP_1 como si él fuese el destino del paquete de descubrimiento de ruta, el cual no tendrá nodos intermedios ya que la conexión entre A y B es directa, y lo firmará.

El paquete REP_2, lo genera con la ruta del paquete que empleó para guardar la ruta almacenada en su tabla de rutas cacheadas. En este caso concreto, ese paquete es un *REP* que se inició en el nodo D y cuyo destinatario era el propio nodo B. Como vemos, el nodo C se encuentra dentro de la ruta que conecta los nodos B y D. Por lo tanto, la ruta de nodos que indicaremos en REP_2 será C−D y los nodos origen y destino asociados serán D y B. Como el nodo origen coincide con el último de los saltos indicados, el nodo A será capaz de distinguir que este paquete que emplea una ruta cacheada se ha llevado a cabo a través de un paquete *REP* y no un *RR*.

Una vez que ya tiene los dos paquetes generados, el nodo B agrega la firma generada para el paquete REP_1 con la que tenía almacenada en la tabla de rutas, correspondiente a REP_2. En este momento ya será capaz de enviar el mensaje de respuesta al nodo A:

$$B \rightarrow A\text{: REPC} \parallel [\textit{Firma de REPC}] \parallel N_B \parallel IP_D \parallel IP_B$$

El nodo A será capaz con los datos facilitados de reconstruir los mensajes, y cada uno de los mensajes firmados por los nodos intermedios que componen la ruta para así verificar la ruta final. Es decir, será capaz de reconstruir una ruta segura desde A hasta el nodo intermedio B gracias a la primera parte del paquete *REPC*. Y después podrá conocer una ruta que una ese nodo intermedio B hasta el nodo destino C, gracias a la segunda parte del paquete *REPC*.

Como la ruta que almacenará la habrá descubierto gracias a la utilización de la ruta cacheada de otro nodo, deberá marcar el bitC de su tabla de rutas para no emplear esta ruta como respuesta a un *RR*.

En algunos casos, como en el del ejemplo, la ruta cacheada que permite a un nodo alcanzar el destino requerido, contiene más información de la solicitada. En este caso, el nodo A obtendrá también una ruta hasta el nodo D que podrá añadir a su tabla de rutas, marcando el bitC que indicará que conoce esa ruta como respuesta de una ruta cacheada.

IV. Conclusion

El empleo de las rutas cacheadas para responder a las solicitudes de descubrimiento de ruta, permite disminuir el número de paquetes que deben ser transmitidos por la red para lograr una ruta válida, sin disminuir por ello la seguridad alcanzada en [11]. Esto se ha conseguido utilizando las opciones y los tipos de paquetes descritos en el RFC que define el protocolo DSR [12]. Como único añadido se requiere el empleo de un bit de la zona reservada dentro del paquete de respuesta *REP*.

Además, el empleo de las rutas cacheadas ha dado como resultado que un nodo pueda aprender más rutas que la esperada y añadir más información a su tabla de rutas.

Como futuras mejoras se plantea que en los paquetes de descubrimiento de ruta, habrá que establecer un método, normalmente mediante la activación de un bit en el *RR*, que permita paquetes *REP* formados con rutas cacheadas.

El nivel de seguridad no varía de la anterior propuesta, ya que se emplea la misma primitiva criptográfica, sobre otro tipo de datos, que permiten ser verificados en una extensión mayor que la original. La única posibilidad que se vislumbra en el ataque es la posible inclusión de rutas caducadas, para lo que sería necesario establecer algún tipo de sincronización y sellado en el tiempo que será motivo de posteriores estudios.

Agradecimientos

Este trabajo ha sido subvencionado por la Cátedra Telefónica de la Universidad de Zaragoza.

Referencias

[1] Perkins, C.E., Royer, E.M.: “Ad-hoc on-demand distance vector routing´´, en WMCSA ’99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, IEEE Computer Society (1999) 90–100

[2] Johnson, D.B., Maltz, D.A.: “Dynamic source routing in ad hoc wireless networks´´, en Imielinski, Korth, eds.: Mobile Computing. Volume 353. Kluwer Academic Publishers (1996)

[3] Perkins, C.E., Bhagwat, P.: “Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers´´, en SIGCOMM ’94: Proceedings of the conference on Communications architectures, protocols and applications, ACM Press (1994) 234–244

[4] Murthy, S., Garcia-Luna-Aceves, J.J.:“ An efficient routing protocol for wireless networks´´. Mob. Netw. Appl. 1(2) (1996) 183–197

[5] Park, V.D., Corson, M.S.:“A highly adaptive distributed routing algorithm for mobile wireless networks´´, en INFOCOM ’97: Proceedings of the INFOCOM ’97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution, IEEE Computer Society (1997) 1405

[6] Toh, C.K.:“A novel distributed routing protocol to support ad-hoc mobile computing´´,en Proceedings of 15 IEEE Annual International Phenix Conference on Computers and Communications. (1996) 480–486

[7] Zapata, M.G., Asokan, N.: “Securing ad hoc routing protocols´´,en WiSE ’02: Proceedings of the 3rd ACM workshop on Wireless security, ACM Press (2002) 1–10

[8] Sanzgiri, K., Dahill, B., Levine, B.N., Shields, C., Belding-Royer, E.M.: “ A secure routing protocol for ad hoc networks´´, en ICNP ’02: Proceedings of the 10th IEEE International Conference on Network Protocols, IEEE Computer Society (2002) 78–89

[9] Hu, Y.C., Perrig, A., Johnson, D.B.: Ariadne: “A secure on-demand routing protocol for ad hoc networks´´, en MobiCom ’02: Proceedings of the 8th annual international conference on Mobile computing and networking, ACM Press (2002) 12–23

[10] Kim, J., Tsudik, G.: Srdp: “Securing route discovery in dsr´´, en The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services. (2005) 247–260

[11] Piles,J.J, Salazar, J.L.: “Encaminamiento seguro para redes ad-hoc´´, en IX Reunión Española sobre criptología y seguridad de la información (RECSI 2006) pp. 732–744. Septiembre. 2006.

[12] David B. Johnson, Yih-Chun Hu, and David A. Maltz, ”The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4”, Internet Request for Comments RFC 4728, February 2007.

[13] Okamoto, T.: “A digital multisignature scheme using bijective public-key cryptosystems´´, en ACM Trans. Comput. Syst. 6(4) (1988) 432–441

[14] Boneh, D., Gentry, C., Lynn, B., Shacham, H.: “Aggregate and verifiably encrypted signatures from bilinear maps´´, en Cryptology ePrint Archive, Report 2002/175. Volume 2656 of Lecture Notes in Computer Science. (2002) 416–432

[15] Lysyanskaya, A., Micali, S., Reyzin, L., Shacham, H.: “Sequential aggregate signatures from trapdoor permutations´´, en Proceedings of Eurocrypt 2004. Volume 3027 of Lecture Notes on Computer Science. (2004) 74–90

[16] Herranz, J.: “Deterministic identity-based signatures for partial aggregation´´, en The Computer Journal 49(3) (2006) 322–330

[17] Crépeau, C., Davis, C.R.: “A certificate revocation scheme for wireless ad hoc networks´´, en SASN ’03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, ACM Press (2003) 54–61

[18] Salazar, J.L., Ruíz, J., Gallardo, P.: “Desarrollo de un entorno seguro de comunicación en una red adhoc´´, en RECSI ’04: VIII Reunión Española sobre Criptología y Seguridad de la Información. (2004) 447–454

[19] Luo, J., Hubaux, J.P., Eugster, P.T.: “DICTATE: DIstributed CerTification Authority with probabilisTic frEshness for Ad Hoc Networks´´, en IEEE Transactions on Dependable and Secure Computing 2(4) (2005) 311–323

Seguridad en protocolos de encaminamiento para redes DTN

Sergio Castillo-Pérez, Sergi Robles, MCarmen de Toro, and Joan Borrell
Department of Information and Communications Engineering (dEIC)
Universitat Autònoma de Barcelona
08193 Bellaterra, Spain
Email: {Sergio.Castillo, Sergi.Robles, MariaCarmen.deToro, Joan.Borrell}@uab.cat

***Resumen*—Las redes tolerantes a retrasos e interrupciones (DTN) no admiten los protocolos de la familia TCP/IP para el transporte de información ni para realizar un encaminamiento eficiente. Los nuevos protocolos de encaminamiento que van apareciendo para este tipo de redes no tienen en cuenta la estrecha relación entre la aplicación y la propia red. En este artículo se exploran los problemas del encaminamiento en redes DTN, haciendo especial hincapié en sus aspectos de seguridad, y se reflexiona sobre cómo ha de ser su diseño. Un elemento clave será la implicación directa de la aplicación.**

I. Introducción

Las redes basadas en la familia de protocolos TCP/IP, es decir, las internet, han sido aceptadas como un estándar *de facto* para las comunicaciones locales y globales de propósito general. Para este tipo de redes se han definido a lo largo de su existencia una multitud de protocolos subsidiarios que complementan sus funcionalidades, añadiendo desde maneras eficientes para el intercambio de información de encaminamiento, hasta esquemas de protección e intercambio de claves. Incluso después de la llegada de la llamada inteligencia ambiental, cuando las redes ad-hoc móviles han vivido su época de apogeo, los protocolos TCP/IP han continuado en la brecha, aceptando nuevos mecanismos para el descubrimiento de vecinos, para la configuración inicial de parámetros de conectividad, o para el encaminamiento energéticamente eficiente, por citar sólo algunos.

Recientemente, sin embargo, ha aflorado un tipo de redes en las que los ya tradicionales protocolos de internet no pueden usarse. En estas redes, llamadas generalmente redes con dificultades (*challenged networks*), existen requisitos de comunicación que TCP/IP no puede cumplir, como por ejemplo unos tiempos de intercambio de información entre los nodos encaminadores superiores a los máximos previstos (IP, 255 segundos; TCP, 120 segundos). Encontramos redes de este tipo en escenarios muy diversos, como las denominadas redes exóticas (e.g.: comunicaciones espaciales de grandes distancias, o comunicaciones submarinas basadas en ultrasonidos), o en situaciones en las que no existe una comunicación viable a corto plazo con un coste razonable. Actualmente encontramos estas redes bajo el epígrafe de Redes Tolerantes a Retrasos e Interrupciones (en inglés, *Delay and Disruption Tolerant Networking*, o DTN).

Un grupo creado por el *Internet Research Task Force* (IRTF) y denominado *Delay Tolerant Networking Research Group* (DTNRG [10]), ha realizado avances en la definición de protocolos para este tipo de redes. Así, se ha creado el *Bundle Protocol* [28], o el *Licklider Transmission Protocol* [25], que superan la restricciones y limitaciones originales de IP. Estos protocolos están recogidos ya en sendos documentos RFC (*Request For Comments*), por lo que ya están en el camino de convertirse en estándares que complementarán los de la familia TCP/IP. Asimismo, algunos proyectos de investigación están realizando contribuciones en este ámbito, como es el caso de *Haggle* [9]. En paralelo, se han desarrollado otros mecanismos capaces de superar las limitaciones anteriormente mencionadas, como los basados en Agentes Móviles.

Una de las mayores dificultades que retrasa (o bajo la opinión de los autores, incluso imposibilita) la aparición de una familia completa de protocolos de propósito general que dé cobertura a las necesidades de las DTN, estriba en la estrecha relación entre la aplicación que usa la red y la propia manera de operar de dicha red.

En este artículo introducimos el problema del encaminamiento en redes DTN desde la perspectiva de su seguridad, realizando un recorrido desde la evolución de las propuestas más clásicas a los protocolos y tendencias más actuales. Después de analizar el origen de las dificultades encontradas, se cuestionan las líneas actuales de desarrollo de protocolos de este tipo. Sin pretender solucionar este ambicioso problema, este trabajo ofrece unas indicaciones hacia un cambio en el planteamiento actual que permitirán hacer frente a las necesidades reales de encaminamiento seguro de las aplicaciones sobre DTN. La clave del proceso ha de estar en la delegación a las aplicaciones de las funciones de encaminamiento e intercambio de la información necesaria. Sólo de esta manera las DTN podrán transportar con eficiencia la información, sin caer en la trampa de querer encontrar en la capa de red una información inherente que permita tomar decisiones de encaminamiento.

En el resto del artículo se introducen los protocolos de encaminamiento para redes DTN (sección II) y las soluciones de seguridad para los mismos (sección III). Posteriormente, se discute la situación actual y se plantean las tesis de los autores sobre la evolución de los protocolos seguros de encaminamiento para redes DTN, basándose principalmente en sus experiencias en aplicaciones médicas y aeronáuticas para este tipo de redes.

II. Protocolos de encaminamiento en redes DTN

A diferencia de la red Internet, donde los protocolos de encaminamiento están bien definidos (i.e., BGP [26], RIP [22] u OSPF [23]), las redes DTN no disponen en la actualidad de un estándar ampliamente reconocido. A pesar de los esfuerzos realizados por el DTNRG, el cual ha elaborado dos borradores con propuestas diferentes ([8], [19]), la comunidad científica continua realizando aportaciones con alternativas dispares entre ellas. Parece no existir un consenso en cuanto a una estrategia lo suficiente genérica y válida para cualquier casuística.

Seguidamente, en esta sección recogemos diversas taxonomías genéricas de los protocolos de encaminamiento según los criterios de varios autores. El objetivo es poner de manifiesto la gran cantidad de propuestas realizadas. Desde el punto de vista de los autores, la causa de esta disgregación de protocolos viene motivada por las características naturales de las redes DTN y sus aplicaciones. Es decir, el modelo de encaminamiento es fuertemente dependiente de la aplicaciones sobre DTN.

II-A. Esquemas de encaminamiento según el conocimiento de la topología de la red

Podemos encontrar una clasificación de los protocolos de encaminamiento en [11], la cual tiene en cuenta el nivel de conocimiento sobre las características de la topología de la red y la demanda de tráfico que tienen los nodos. Farrell, según este criterio, identifica los siguientes tipos de encaminamiento:

- **Basado en oráculo**: un nodo o un conjunto de nodos (oráculos) tiene un conocimiento casi total de la red y de cómo evolucionará. Este conocimiento es utilizado por los oráculos para distribuir información de encaminamiento en la red que será empleada para determinar la ruta.
- **Basado en modelo:** los algoritmos de encaminamiento utilizan perfiles de comportamiento conocidos con el objetivo de seleccionar las rutas.
- **Epidémico**: basado en la propagación masiva de la información entre todos los nodos (*flooding*), sin tener un conocimiento preciso de cuál será la ruta final.
- **Basado en estimación**: estos algoritmos priorizan las rutas estimando probabilidades, de manera que maximizan el reenvío de un paquete hacia un nodo en particular sabiendo que así se aumentará la probabilidad de entregar el paquete al destino final.
- **Erasure coding**: estos protocolos de encaminamiento emplean estrategias basadas en la teoría de códigos, concretamente los *erasure codes* [21]. Se fundamentan en transformar un mensaje de n símbolos en uno de longitud mayor con k símbolos, de manera que un subconjunto de los k símbolos permite reconstruir el mensaje original. Los algoritmos de encaminamiento son los responsables de crear y enviar dichos k símbolos, con el objetivo de que un subconjunto permita recuperar el mensaje original en el destino.
- **Basado en control del movimiento de los nodos**: a esta categoría pertenecen aquellos esquemas en los que el algoritmo de encaminamiento es utilizado para controlar el movimiento físico de los nodos.

II-B. Estrategias de encaminamiento basado en la propagación de los mensajes

De acuerdo con Balasubramanian y Nelson podemos considerar la siguiente taxonomía basada en las réplicas de un mensaje ([4], [24]):

- **Basado en reenvío:** mantiene una única copia de un mensaje en la red e intenta reenviarla hacia el destinatario en cada encuentro. Algunos de los protocolos que se engloban en esta categoría los encontramos en [16], [17] y [31].
- **Basado en replicación:** inserta múltiples copias (o réplicas) de un mensaje en la red para incrementar la probabilidad de entrega. Los protocolos basados en replicación intentan balancear el compromiso entre los recursos utilizados y la probabilidad de entregar un mensaje. Los protocolos propuestos intentan limitar la replicación o eliminar las réplicas no útiles según diversos criterios como:
  - Utilizar información histórica de encuentros.
  - Eliminar réplicas utilizando paquetes de confirmación de mensajes entregados.
  - Utilizar información probabilística de movilidad para inferir las entregas.
  - Replicar mensajes con una probabilidad baja.
  - Emplear códigos con redundancia.
  - Establecer un número máximo de réplicas.

  En la categoría de replicación podemos identificar dos subcategorías:
  - **Basado en *flooding*:** envía una replica de cada mensaje a tantos nodos como sea posible. Algunos protocolos propuestos son Epidemic [36], PRoPHET [20], MaxProp [7] o RAPID [4].
  - **Basado en cuota:** limita el número de replicas. Algunos de los algoritmos que pertenecen a esta categoría son *Spray and Wait* [32], *Spray and Focus* [30] o *Encounter-Based* [24].
- **Estrategia híbrida:** combina conceptos de encaminamiento basado en reenvío y replicación. A esta categoría pertenece el *Hybrid Probabilistic Routing Scheme Using Multi-Copies (HUM)* [18].

II-C. Esquemas de encaminamiento según accidentalidad o intencionalidad

En [4] podemos encontrar otra categorización basada en los conceptos de accidentalidad o intencionalidad respecto a una cierta métrica de rendimiento. Alguna de estas métricas pueden ser el tiempo promedio de retraso o la probabilidad de entrega entre otros.

- **Accidentalidad**: esquemas que sólo tienen un efecto de mejora fortuito sobre una métrica de rendimiento.

- **Intencional**: esquemas que de forma intencionada pretenden optimizar una métrica de rendimiento escogida.

II-D. Encaminamiento basado en constricciones de recursos

Balasubramanian *et al.* proponen en [4] una taxonomía de los algoritmos de encaminamiento considerando si éstos tienen en cuenta unas constricciones de recursos o no. Entre dichas constricciones podemos encontrar la capacidad de almacenamiento de los nodos, el ancho de banda o el consumo energético.

III. Seguridad en redes DTN

En la actualidad disponemos de un amplio conjunto de mecanismos que nos proporcionan seguridad en las redes. Sin embargo, estos mecanismos tradicionales, como son los protocolos criptográficos SSL o TLS, no pueden ser trasladados a las redes DTN de forma directa. El motivo de esto radica en el hecho de que dichos mecanismos parten de una serie de asunciones no propias de las redes tolerantes a retrasos, tales como una conectividad permanente punto a punto, o retrasos pequeños a nivel de capa de enlace entre otros. Por este motivo, nuevos protocolos y arquitecturas deben ser diseñadas basándose en las características inherentes a las redes DTN. De acuerdo con esto, en la presente sección analizaremos la seguridad desde dos perspectivas relevantes: la gestión de claves criptográficas y el encaminamiento seguro.

III-A. Gestión de claves criptográficas

Actualmente, el *Delay Tolerant Networking Research Group* está definiendo las especificaciones necesarias para proporcionar seguridad en las redes DTN. Concretamente, se han publicado dos documentos en forma de RFC y de borrador para el *Licklider Transmission Protocol* (LTP) y para el *Bundle Protocol* (BP) respectivamente.

En el caso del LTP, la seguridad viene definida a través del RFC 5327 [13], donde se plasman las extensiones de seguridad necesarias para garantizar la autenticidad de los fragmentos. Análogamente, se propone evitar posibles ataques DoS mediante la utilización de valores aleatorios –denominados *cookies*– que son añadidos a las cabeceras de los fragmentos.

En el caso particular del protocolo *bundle*, se ha presentado un borrador [35] que recoge las especificaciones para proporcionar seguridad. En dicho borrador se definen cuatro bloques que pueden ser añadidos a las cabeceras del protocolo *bundle* con la finalidad de proporcionar diversos servicios de seguridad. De forma concreta, éstos son el *Bundle Authentication Block* (BAB), el *Payload Integrity Block* (PIB), el *Payload Confidentiality Block* (PCB) y el *Extension Security Block* (ESB).

En ambos documentos se define un conjunto de funciones criptográficas a emplear de forma mandatoria, como pueden ser RSA, SHA o AES. Así pues, la criptografía es propuesta como una herramienta a emplear para resolver los problemas característicos de seguridad de confidencialidad, integridad y autenticidad. Sin embargo, tanto en las mismas propuestas del DTNRG como en otras referencias ([2], [3], [11], [1], [29]) se reconoce abiertamente que la distribución y gestión de claves en las redes DTN es un problema abierto y vivo desde la perspectiva de la investigación.

Tradicionalmente, el problema de la gestión de claves ha sido resuelto mediante las infraestructuras de clave pública (PKI - *Public Key Infrastructure*). Sin embargo, esta arquitectura no puede emplearse en entornos tolerantes a desconexiones y retrasos como son las redes DTN. Así, un emisor en una red DTN no siempre dispondrá del certificado digital del destinatario que necesite. Hay que considerar, a su vez, que ante una situación similar tampoco tenemos la garantía que éste pueda acceder a un repositorio de donde obtener el certificado. Del mismo modo, garantizar la autenticidad de los certificados implicaría que cada nodo almacenase un conjunto de certificados asociados a las autoridades de certificación de confianza. Este hecho, y dependiendo del número de certificados, puede conducirnos a una situación inadmisible motivada por las restricciones de capacidad de almacenamiento de determinados nodos. De forma similar, podemos considerar que la actualización de los certificados de revocación y su almacenamiento supone una problemática en entornos DTN.

Diversas ideas en este área han sido propuestas hasta la fecha para solventar algunas de las problemáticas vistas anteriormente. Así, tal y como se expone en [12], la adopción del esquema *duckling* —propio de las redes ad-hoc— podría ser una primera aproximación a la solución. Dicho esquema, propuesto en [34] y extendido en [33], propone el intercambio de claves entre nodos que tienen una alta probabilidad de conectividad con el resto, almacenándolas y reenviándolas a otros nodos en futuras conexiones oportunistas. A pesar de esto, la adopción de este esquema no resuelve el problema en su totalidad debido a su no-determinismo, ya que no garantiza que los nodos dispongan de todas las claves públicas en cierto instante de tiempo. De forma similar, la autenticidad de las claves reenviadas es un problema no resuelto por este esquema. Podemos notar que la revocación de claves tampoco es solventada, ya que el envío de las listas de actualizaciones con las claves revocadas podría postergarse excesivamente en un contexto de desconectividad como son las redes DTN.

Una alternativa para resolver los problemas anteriores se fundamenta en emplear claves basadas en la identidad de los nodos. Esta idea, conocida como *Identity Cryptography-Based* (IBC [5]), es propuesta por Asokan *et al.* para DTN en [2] y en [3]. A diferencia de las PKI tradicionales, donde un usuario obtiene las claves de una autoridad de certificación, en IBC la clave pública es construida a partir de un identificador vinculante a cada nodo, mientras que la clave privada es generada por una tercera parte de confianza denominada *Private Key Generator* (PKG). De esta manera, para un emisor, la obtención de una clave pública de un destinatario se transforma en un proceso trivial y carente de cualquier comunicación con repositorios de claves. El problema de la revocación de claves es resuelto mediante la construcción de identificadores (i.e., claves públicas) temporales. Así, un identificador es utilizado por un corto periodo de tiempo, y construido mediante la concatenación de un identificador fijado para un nodo y destinatario junto a su periodo de validez (e.g.,

`alice@dtn-node.com:23-04-2010`). Mediante esta estrategia, la revocación de claves es evitada al refrescar los identificadores de forma periódica. Podemos encontrar algunas mejoras sobre el esquema IBC presentadas en [29], y denominada *Hierarchical Identity Based Cryptography* (HIBC).

III-B. Encaminamiento y seguridad en DTN

Desde el punto de vista del encaminamiento, el algoritmo empleado en una red DTN tiene una fuerte influencia en las propiedades de seguridad del sistema. Así, mientras que determinadas estrategias de encaminamiento pueden ser diseñadas sin mecanismos de autenticación, otras podrían necesitar de éstos como condición *sine qua non* para su correcto funcionamiento. Así, por ejemplo, en [6] se plantea cómo una estrategia de encaminamiento adecuada, sin el uso de mecanismos de autenticación entre nodos intermedios, puede llegar a ser efectiva. Asimismo, la utilización o no de autenticación en el proceso de encaminamiento no elude la necesidad de autenticación y/o confidencialidad de extremo a extremo entre aplicaciones. En este sentido, y tal como ocurre hoy en día en la red Internet, podemos notar también una dependencia entre las características de seguridad del sistema y los requisitos de las aplicaciones. Así, podemos pensar que, por ejemplo, determinadas aplicaciones no críticas no tienen porque garantizar la confidencialidad de la información que viaja entre los nodos de la red DTN, mientras que para otras suponga una necesidad indispensable.

Como sugiere Farrell en [11], es necesario que los protocolos y las implementaciones soporten mecanismos de encaminamiento basados en políticas. Es decir, cada protocolo de DTN debería especificar qué variables de seguridad tienen que ser consideradas desde el punto de vista del encaminamiento, de manera que las implementaciones puedan tomar decisiones en cuanto al encaminamiento y el reenvío de *bundles*. Esto cobra especial sentido si tenemos en cuenta que el reenvío o almacenaje de un *bundle* supone un consumo de recursos. En particular, dado que el reenvío de un *bundle* implica un gasto de recursos (en términos de espacio, energía, etc...), un nodo debería ser capaz de incorporar una política de encaminamiento que le permitiese tomar una decisión al respecto. Así, por ejemplo, un nodo intermedio podría exigir que todos los *bundles* de entrada debieran ser autenticados, en caso contrario éstos serían rechazados.

IV. Hacia un encaminamiento seguro orientado a la aplicación

Según hemos visto hasta ahora, pese a disponer ya de diversos mecanismos para el transporte de la información y su encaminamiento en redes DTN, aun no es posible aprovechar estas redes en todas sus posibilidades. El problema en si tiene dos vertientes: por una parte, necesitamos que las unidades de información puedan ser dirigidas hacia los nodos que ofrezcan una mayor probabilidad de llegada al destino bajo las restricciones impuestas por la red y la aplicación; por otro lado, precisamos de esquemas de seguridad que sean robustos a la no contemporaneidad de las comunicaciones (los extremos pueden no estar conectados a la red simultáneamente), propia de las DTN. Ni lo uno ni lo otro encuentra en el estado del arte actual unas soluciones viables, pero analicemos aquí si existen otras tecnologías o líneas de investigación que cumplan para con estos propósitos.

IV-A. Encaminamiento

Veamos primero el caso del encaminamiento. Un punto importante que no debe olvidarse aquí es que, a diferencia de las redes *ad-hoc* móviles (MANET), el mejor nodo hacia el cual re-encaminar la información no es necesariamente el que está a menor distancia (de hecho, podemos suponer que la mayoría de los destinos no está accesible en ese preciso instante); ni siquiera, como sucedía usualmente en las redes de sensores inalámbricas, al que está a menor distancia física. Sólo las aplicaciones saben decidir el mejor camino ante un conjunto de vecinos. Por tanto, la no disponibilidad simultanea de las partes comunicantes, combinada con la movilidad de los nodos encaminadores (*store, carry and forward*), hacen que sea imposible predecir a nivel de red qué vecino es el mejor candidato para re-encaminarle la información.

La solución más evidente es dejar decidir a la propia información. Es decir, la unidad básica de transmisión puede llevar código implementando el algoritmo de encaminamiento. También, como no, para la actualización de la información acerca de nodos remotos que utilizará ese algoritmo. El tipo de encaminamiento va a depender, no ya de las características de la red subyacente, sino de la propia aplicación. Las aplicaciones utilizan la red de forma muy diferente dependiendo de sus objetivos particulares. Un ejemplo sería una red DTN constituida por autobuses urbanos. Para una aplicación, el algoritmo de encaminamiento podría utilizar el identificador de la línea de autobús de cada vecino, ya que podría conocer *a priori* los enlaces y por tanto decidir de manera correcta; otra aplicación, en el mismo instante, podría utilizar un criterio diferente, como si aquel autobús en concreto ya ha sido visitado anteriormente por el agente.

El código móvil, y particularmente los agentes móviles, son una alternativa clara a los esquemas actuales de encaminamiento en redes DTN, y permiten implementar las ideas expuestas. En este caso, debe considerarse que existe un doble transporte de la información: el de los nodos físicamente móviles, y el de los agentes, que transportarían los datos a distancias de uno o más nodos. Así, varias aplicaciones darían lugar a diferentes tipos de agentes, cada uno con sus propios algoritmos de encaminamiento operando simultáneamente. Para el intercambio de información de encaminamiento se necesitaría una estructura más compleja, que podría consistir en una organización basada en ontologías.

IV-B. Seguridad

Una vez vistas las consideraciones sobre el encaminamiento, enlazamos con la cuestión prioritaria: la seguridad. En la sección III hemos visto los diferentes esquemas de seguridad que se han planteado hasta el momento relacionados con las redes DTN. Hemos concluido que ninguno de los esquemas

propuestos constituye una solución válida de propósito general, puesto que al final la mayoría presupone la existencia y accesibilidad de algún componente generalmente utilizado para la gestión de claves. Ésto, en una red DTN, plantea un requisito difícilmente alcanzable.

Por tanto, no nos sirven para las DTN los esquemas de seguridad que se basan en el establecimiento eventual de una conexión a un punto centralizado. Una propuesta interesante es la denominada criptografía basada en la identidad, o IBC, donde una parte de confianza genera las claves privadas que corresponden a identificadores, que hacen a su vez de clave pública. IBC, combinada con [35] sí plantea un escenario viable para el uso de criptografía de clave publica en escenarios DTN. No obstante, hay que tener en cuenta varios aspectos. En primer lugar, IBC añade en las identificaciones/claves públicas información sobre el periodo de validez de la clave con el claro objetivo de no requerir un sistema de revocación de certificados basado en el intercambio de listas. Para su utilización en DTN debe considerarse que los nodos, dependiendo de la red, pueden estar días o incluso meses desconectados y que, por lo tanto, los periodos de validez deberían ser largos. Esto puede afectar a la robustez del esquema, dependiendo de la IBC concreta utilizada. El acceso al productor de claves (PKG) en IBC en ciertos escenarios DTN puede llegar a ser también un problema insalvable.

Los agentes móviles también aportan una alternativa para afrontar el problema de la seguridad de la información (de encaminamiento o no) en las redes DTN. Existen propuestas, como [14] y [15], que dotan a los agentes de la capacidad de decidir y aplicar sus propios esquemas de seguridad, al margen de los entornos de ejecución. En este caso, las unidades de información están auto-protegidas.

Cuando por las especificidades de la aplicación o la red no pueda usarse ninguno de los enfoques clásicos, ni de los mencionados hasta ahora, también se puede recurrir a esquemas basados en modelos de confianza [27]. La reputación, por ejemplo, ganada por los nodos en un histórico reciente de interacciones, podría usarse de manera combinada con un sistema dinámico de recomendaciones para valorar el riesgo de atentados contra la seguridad de las comunicaciones, o para aplicar técnicas de ostracismo a los nodos problemáticos.

V. Conclusiones

Las redes DTN presentan unos requisitos nada fáciles de cumplir, tanto por lo que se refiere al transporte de la información, como a su encaminamiento, y especialmente en cuanto a su seguridad. A lo largo de este artículo se han explorado los diferentes mecanismos que existen hoy en día para hacer frente a todos estos problemas, con el objetivo final de determinar si una combinación de los mismos es suficiente o si por el contrario debe plantearse otro tipo de soluciones.

Después de la revisión de mecanismos realizada en las primeras secciones de este trabajo, y de las reflexiones en la sección anterior, una de las primeras y más claras conclusiones que extraemos es que las líneas de investigación basadas en un enfoque clásico, continuista de los protocolos de Internet, no solucionarán los problemas en las DTN. En este tipo de redes, las aplicaciones adquieren un alto nivel de protagonismo que en muchos casos pasan a ser ellas las únicas capacitadas para la toma de decisiones.

Unas de las maneras más evidentes de dotar a las aplicaciones de estas capacidades, sin mezclar la implementación de la funcionalidad principal con la de la gestión de la red, es a través del uso de agentes móviles. Esta manera de hacer se desmarca completamente de la seguida hasta el momento por el DTNRG, que siguen el modelo de IP hasta donde es posible y quedan bloqueados cuando hay que aportar una solución a los problemas más duros. Los agentes móviles pueden ser el impulso necesario para salir de esta solución de máximo local.

Muchas de las soluciones concretas que hemos revisado implican que todos los elementos de la red deben usarlos del mismo modo, por ejemplo un determinado algoritmo de encaminamiento. En este aspecto, los agentes móviles nos permiten una convivencia entre diferentes esquemas, sin que se interfieran entre ellos y facilitando enormemente las tareas de mantenimiento (como la corrección de errores).

Los agentes móviles nos ofrecen una manera completa de tener mecanismos de encaminamiento en redes DTN con aplicaciones heterogéneas, y al mismo tiempo posibilitan añadir seguridad, y por extensión, encaminamiento seguro que es lo que buscábamos en un principio.

No estamos ante una solución ya preparada para solventar todos los problemas que plantean las DTN. Para poder utilizar los agentes móviles en DTN se deben integrar mecanismos en los mismos que diferencien la ejecución autónoma que los caracteriza, con otra no autónoma, de soporte de decisión de encaminamiento, o de actualización de información de encaminamiento. Para realizar todo esto se necesita un planificador del encaminamiento para agentes móviles en DTN, que podría incorporar también una gestión de prioridades.

Al margen de los agentes móviles, y cuando por cuestiones de la aplicación o de la red no puedan usarse los esquemas revisados, puede recurrirse a una solución basada en la confianza. Esquemas de reputación y recomendación serían mecanismos que permitirían, después de un tiempo transitorio, disponer de una protección tácita contra ataques.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el Ministerio de Industria, Turismo y Comercio, a través del proyecto de referencia TSI-020100-2009-115.

Referencias

[1] K. Aniket, Z. Gregory M., and H. Urs. Anonymity and Security in Delay Tolerant Networks. In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pages 504–513, 2007.

[2] N. Asokan, K. Kostianinen, P. Ginzboorg, J. Ott, and C. Luo. Applicability of Identity-Based Cryptography for Disruption-Tolerant Networking. In *MobiOpp '07: Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, pages 52–56, New York, NY, USA, 2007. ACM.

[3] N. Asokan, K. Kostianinen, P. Ginzboorg, J. Ott, and C. Luo. Towards Securing Disruption-Tolerant Networking. *Nokia Research Center, Tech. Rep. NRC-TR-2007-007*, 2007.

[4] A. Balasubramanian, B. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, page 384. ACM, 2007.

[5] D. Boneh and M. K. Franklin. Identity-Based Encryption from the Weil Pairing. In *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 213–229, London, UK, 2001. Springer-Verlag.

[6] J. Burgess, G. D. Bissias, M. D. Corner, and B. N. Levine. Surviving Attacks on Disruption-Tolerant Networks Without Authentication. In *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 61–70, New York, NY, USA, 2007. ACM.

[7] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–11, 2006.

[8] S. Burleigh. Contact Graph Routing. Internet-Draft (Experimental), draft-burleigh-dtnrg-cgr-00, December 2009.

[9] C. Diot and *et al.* Haggle project. http://www.haggleproject.org/, January 2006.

[10] DTNRG. Delay Tolerant Networking Research Group. http://www.dtnrg.org.

[11] S. Farrell and V. Cahill. *Delay- and Disruption-Tolerant Networking*. Artech House, Inc., Norwood, MA, USA, 2006.

[12] S. Farrell and V. Cahill. Security Considerations in Space and Delay Tolerant Networks. In *SMC-IT '06: Proceedings of the 2nd IEEE International Conference on Space Mission Challenges for Information Technology*, pages 29–38, Washington, DC, USA, 2006. IEEE Computer Society.

[13] S. Farrell, M. Ramadas, and S. Burleigh. Licklider Transmission Protocol - Security Extensions. RFC 5327 (Experimental), September 2008.

[14] C. Garrigues, S. Robles, and J. Borrell. Securing Dynamic Itineraries for Mobile Agent Applications. *Journal of Network and Computer Applications*, 31(4):487–508, November 2008.

[15] C. Garrigues, S. Robles, J. Borrell, and G. Navarro-Arribas. Promoting the development of secure mobile agent applications. *Journal of Systems and Software*, In Press, Corrected Proof:–, 2009.

[16] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Network. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 145–158, New York, NY, USA, 2004. ACM.

[17] Evan P. C. Jones, Lily Li, and Paul A. S. Ward. Practical Routing in Delay-Tolerant Networks. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 237–243, New York, NY, USA, 2005. ACM.

[18] Z. Li and H. Shen. Probabilistic Routing with Multi-Copies in Delay Tolerant Networks. In *Distributed Computing Systems Workshops, 2008. ICDCS '08. 28th International Conference on*, pages 471–476, June 2008.

[19] A. Lindgren, A. Doria, E. Davies, and S. Grasic. Probabilistic Routing Protocol for Intermittently Connected Networks. Internet-Draft (Experimental), draft-irtf-dtnrg-prophet-05, February 2010.

[20] A. Lindgren, A. Doria, and O. Schelén. Probabilistic Routing in Intermittently Connected Networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, 2003.

[21] M. Luby, M. Mitzenmacher, M.A. Shokrollahi, and D.A. Spielman. Efficient Erasure Correcting Codes. *IEEE Transactions on Information Theory*, 47(2):569–584, 2001.

[22] G. Malkin. RIP Version 2. RFC 2453 (Standard), November 1998. Updated by RFC 4822.

[23] J. Moy. OSPF Version 2. RFC 2328 (Standard), April 1998. Updated by RFC 5709.

[24] S. Nelson, M. Bakht, and R. Kravets. Encounter-Based Routing in DTNs. In *Proceedings of INFOCOM 2009*, April 2009.

[25] M. Ramadas, S. Burleigh, and S. Farrell. Licklider Transmission Protocol - Specification. RFC 5326 (Experimental), September 2008.

[26] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), January 2006.

[27] J. Sabater and C. Sierra. Review on Computational Trust and Reputation Models. *Artif. Intell. Rev.*, 24(1):33–60, 2005.

[28] K. Scott and S. Burleigh. Bundle Protocol Specification. RFC 5050 (Experimental), November 2007.

[29] A. Seth and S. Keshav. Practical Security for Disconnected Nodes. In *1st IEEE ICNP Workshop on Secure Network Protocols, 2005.(NPSec)*, pages 31–36, 2005.

[30] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility. In *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, pages 79–85, March 2007.

[31] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Single-copy Routing in Intermittently Connected Mobile Networks. In *Proceedings of IEEE SECON*, pages 235–244, 2004.

[32] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Spray and Wait: an Efficient Routing Scheme for Intermittently Connected Mobile Networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, page 259. ACM, 2005.

[33] F. Stajano. The Resurrecting Duckling - What Next? In *Revised Papers from the 8th International Workshop on Security Protocols*, pages 204–214, London, UK, 2001. Springer-Verlag.

[34] F. Stajano and R. J. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *Proceedings of the 7th International Workshop on Security Protocols*, pages 172–194, London, UK, 2000. Springer-Verlag.

[35] S. Symington, S. Farrell, H. Weiss, and P. Lovell. Bundle Security Protocol Specification. Internet-Draft (work in progress), draft-irtf-dtnrg-bundle-security-15, February 2010.

[36] A. Vahdat and D. Becker. Epidemic Routing for Partially Connected ad hoc Networks. Technical report, Duke University, 2000.

Seguridad en la planificación de agentes móviles en redes DTN

Carlos Borrego
High Energy Physics Institute (Ifae)
Universitat Autònoma de Barcelona
08193 Bellaterra, Spain
Email: cborrego@ifae.es

Sergi Robles
Department of Information and
Communications Engineering (dEIC)
Universitat Autònoma de Barcelona
08193 Bellaterra, Spain
Email: sergi.robles@uab.cat

***Abstract*—Los protocolos de almacenamiento, transporte y reenvío, *store, carry and forward*, para redes DTN, redes tolerantes a retrasos, y en particular el protocolo *Bundle protocol*, ofrecen nuevas posibilidades en escenarios donde la interconexión es intermitente, los anchos de banda son asimétricos, las latencias son altas y variables y los patrones de movilidad son ambiguos. En el contexto del protocolo *Bundle protocol*, sus unidades de protocolo, los *bundles*, permanecen acumulados durante un tiempo en los nodos DTN hasta que otro nodo acepte su custodia. En algunos escenarios DTN, este tiempo puede ser inadmisiblemente elevado. *Bundles* considerados cruciales pueden permanecer bloqueados por mucho tiempo, mientras otros menos importantes son liberados. Por otro lado, los agentes móviles son un modo excelente de implementar redes DTN ya que pueden transportar los *bundles* o directamente información de aplicación y a la vez ejecutar código. Las plataformas encargadas de permitir la ejecución de los agentes móviles, conservan éstos hasta que otra plataforma acepte su migración. Las migraciones de los agentes móviles pueden ser planificadas usando políticas dinámicas para evitar que agentes móviles importantes sean bloqueados. Estas políticas pueden viajar con los mismos agentes móviles. Este artículo describe los problemas de seguridad que puede comportar este tipo de escenarios.**

I. Introducción

En los últimos años están apareciendo unos nuevos tipos de redes, las llamadas redes con dificultades (*challenged networks*) para las que los habituales protocolos de Internet, la familia TCP/IP, no son válidos. Los principales motivos son la no contemporaneidad de las comunicaciones, y la prevalencia de las interrupciones, desconexiones, grandes retrasos, e intermitencias, lo que invalida todos aquellos protocolos con un tiempo máximo de espera o vida de las unidades de información, como IP (255 segundos), o TCP (120 segundos). Entre estas redes se encuentran la redes inalámbricas dinámicas, redes de sensores heterogéneas, y las denominadas redes exóticas, que incluyen las submarinas de ultrasonidos y las interplanetarias. Recientemente se está realizando un esfuerzo de investigación sobre estas redes, bajo el epígrafe de Redes Tolerantes a los Retrasos e Interrupciones (en inglés, *Delay and Disruption Tolerant Networking*, DTN), y el IRTF (*Internet Research Task Force*) ha creado un grupo con el objetivo de promover una serie de protocolos que les den cobertura [2].

Las características de las DTN hacen muy difícil, sino imposible, disponer de mecanismos eficientes de encaminamiento usando exclusivamente información de nivel de red. Los Agentes móviles [3] son una alternativa a los recientemente aparecidos protocolos de transporte de información para DTN ([1]), Licklidder Transport Protocol [8]), basados éstos últimos en el enfoque clásico del encaminamiento y continuista en las líneas de TCP/IP. En este tipo de redes, las propias aplicaciones suelen disponer de información relevante para el encaminamiento, y la habitual separación entre información de las capas de red, transporte y aplicación queda obsoleta. Aquí, los agentes móviles permiten delegar el proceso de encaminamiento (y re-encaminamiento) a la capa de aplicación, permitiendo viajar al mismo tiempo a información y algoritmos de *routing*.

Implementar una red DTN con agentes móviles aplicando las ideas expuestas no es una tarea sencilla. En este articulo proponemos precisamente las bases de un esquema para conseguirlo, centrando en el diseño del planificador de la migración en la plataforma (entorno de ejecución de los agentes móviles), y preservando al máximo la independencia entre la programación de la aplicación y de los algoritmos de encaminamiento. Se hace especial énfasis en su seguridad, ya que los planificadores serán los elementos clave del funcionamiento de la red entera, y por tanto su talón de aquiles.

En el resto del artículo se analizarán las redes DTN basadas en agentes móviles (sección II), y los retos de seguridad asociados al planificador (sección III). Finalmente, se establecen los fundamentos para una arquitectura segura para redes DTN basada en agentes móviles (sección IV).

II. Redes DTN basadas en Agentes Móviles

Las redes DTN tradicionales usan el protocolo *Bundle protocol* [1] para gestionar el paradigma *store-carry-and-forward* (almacenamiento, transporte y reenvío). Como se propone en la figura 1 podríamos complementar la capa *bundle* con una infraestructura basada en agentes móviles.

La información de aplicación viaja usando la infraestructura de las redes DTN representada en la figura como la capa DTN. Esta capa DTN puede ser implementada por un lado usando el protocolo *Bundle protocol* o por otro prescindiendo de éste y usando agentes móviles. Ambas opciones necesitan un nivel de convergencia por debajo, el cual puede ser protocolos basados en protocolos TCP/IP, el protocolo DTN Licklider

Transmission Protocol [8] sobre cualquier nivel de enlace o directamente sobre un nivel de enlace.

Nuestra propuesta consiste precisamente en añadir esta alternativa. El objetivo es conseguir, usando agentes móviles, el paradigma *store-carry-and-forward*. Si suponemos que la plataformas de los agentes móviles pueden estar desconectadas esporádicamente, estamos ante una posible arquitectura DTN. El *almacenamiento* son los agentes móviles mientras que ejecutan su código junto al tiempo de espera bloqueados en la cola de la plataforma. El *transporte* es la misma plataforma en el caso de que sea una plataforma en movimiento. El *reenvío*, en cambio, es la migración del agente móvil. Por tanto, información que viaje dentro de un agente móvil y vaya saltando de plataforma en plataforma, cuando éstas pueden estar esporádicamente desconectadas, es decididamente información del nivel de aplicación de una red DTN.

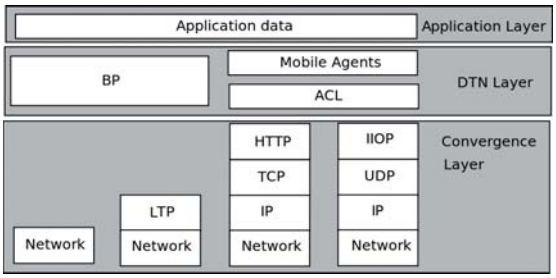


Fig. 1. Capas DTN

En la figura 2 se puede ver como todas estas arquitecturas diferentes planteadas pueden convivir unas con otras. En la misma figura, inicialmente un agente móvil transporta directamente información del nivel de aplicación. El agente ante estas interrupciones, desconexiones, grandes retrasos, e intermitencias, al migrar, se comporta como una *mula* [11] efectuando un *carry* activo, es decir transportando activamente la información del nivel de aplicación. Un nodo custodio DTN en movimiento, como podría ser una barca, un avión o un animal que disponga de una plataforma de agentes móviles, al desplazarse, realiza un *carry* pasivo. Es este caso, no existe ninguna entidad software que efectúe el transporte, sino que es el movimiento físico del nodo custodio quien produce el *carry*. Siguiendo con el flujo de los datos de aplicación de la figura 2, el agente móvil transportado por el nodo custodio en movimiento podría llegar a un escenario en el que ya no se dispone de plataformas de agentes móviles, pero si de la infraestructura clásica del protocolo *Bundle protocol*. En este caso, el agente móvil podría ser *serializado* y viajar dentro de un *bundle*. Una vez en un escenario con plataformas de agentes móviles disponibles, el agente podría ser *revivido* y continuar hacia su destino. Por último, un agente móvil podría incluso querer transportar un bundle. Al llegar a una red DTN tradicional sin plataformas de agentes móviles debería ser capaz de *suicidarse*, sin antes crear una réplica del *bundle* que lleva dentro.

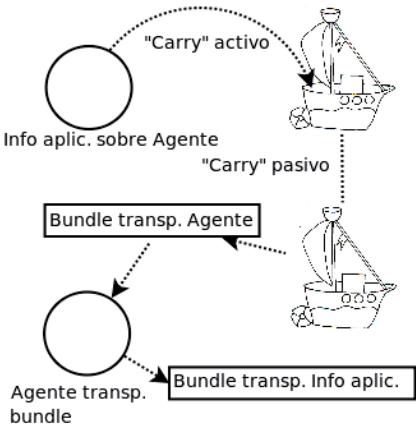


Fig. 2. Distintos escenarios DTN

A. Planificación de agentes móviles

El objetivo de permitir a los *bundles* o a la información de aplicación ser transportados por agentes móviles, es aprovechar la posibilidad de que los agentes móviles son capaces de tomar decisiones en cada uno de los saltos entre nodos custodio. En redes DTN el contexto puede ser modificado lo suficientemente para pensar que se necesita una manera más dinámica de resolver los problemas subyacentes de este tipo de redes. Para poder conseguir este dinamismo, pensamos que la unidad de protocolo *bundle* debería, de alguna manera, poder ejecutar código. Este código ayudará a mejorar aspectos de las redes DTN como el encaminamiento, la congestión, el control de flujo, la fiabilidad y a disminuir los retrasos. Así como el poder ejecutar código nos permitirá más flexibilidad, será la fuente mayor de problemas de seguridad.

El objetivo de planificar las migraciones de los agentes móviles en las plataformas es priorizar determinados agentes móviles sobre otros, dados unos criterios. La necesidad de la priorización se encuentra en el hecho que la plataforma destino donde un agente móvil planea migrar podría estar disponible sólo por un cierto tiempo. Cuanto antes se permita migrar a los agentes móviles más prioritarios, mejor. Por ejemplo, en escenarios de recuperación ante desastres, un agente móvil que contenga información relativa a víctimas de estado grave, debería poder migrar antes que otro agente que contenga información sobre víctimas de estado leve.

La utilidad y la viabilidad de este sistema se ha medido usando tres escenarios diferentes. En los tres escenarios, varios nodos DTN con plataformas corriendo en cada una de ellas, se desconectan de la red esporádicamente y van recibiendo sendos agentes móviles. En las abscisas se representa el número de agentes y en las ordenadas el tiempo de mejora de los agentes priorizados sobe el resto de agentes móviles durante la prueba.

Se define un primer escenario en el que la densidad de agentes móviles es alta sobre una plataforma en concreto, un segundo escenario en el que el patrón de migración es aleatorio y un último escenario en el que, al contrario de los otros dos escenarios, el número de agentes priorizados es alto. Se puede

ver en la figura 3 que a partir de cierto número de agentes móviles, la mejora sobre la media del resto de los agentes es considerable.

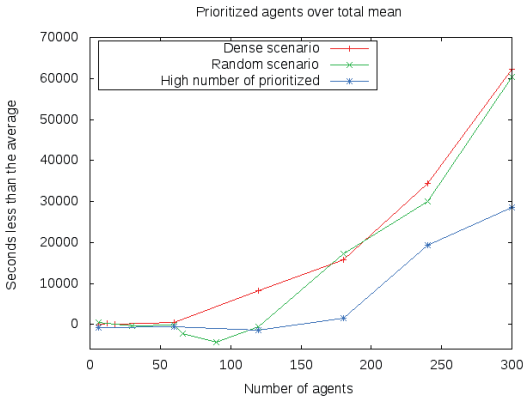


Fig. 3. Agentes móviles priorizados

III. Retos de Seguridad

En esta sección se describirán los problemas de seguridad derivados del uso de planificadores de agentes móviles para escenarios de redes DTN. En la siguiente sección se propondrán soluciones generales para los problemas propuestos.

A. Criterios de planificación

Los criterios para priorizar las migraciones de los agentes móviles podrían cambiar eventualmente en cualquier momento. Complementando el ejemplo de la sección anterior, ante un desastre natural de gran envergadura, los criterios para priorizar agente móviles que transportan información de nivel de aplicación sobre víctimas, pueden variar según las circunstancias.

Para informar a plataformas de criterios nuevos se utilizan los mismos agentes móviles que actúan a modo de *piggybacking* transportando información relativa a las prioridades. Cuando un agente móvil llega a una plataforma, la primera acción que toma es comprobar que la información sobre los criterios de planificación están actualizados. En el caso de que la información que disponga el agente móvil sea más reciente que la disponible localmente en la plataforma, el agente móvil actualizará ésta última. La estructura donde los agentes móviles pueden leer y escribir sobre criterios de prioridad es una estructura local donde, a modo de *pizarra*, los agentes móviles pueden leer y escribir. Los autores ya han experimentado con escenarios similares en los que agentes móviles actualizan información dinámica en nodos heterogéneos en entornos de computación distribuida grid [16].

No podemos confiar en el hecho de que en todo momento la información sobre nuevos criterios de planificación se propague por todos las plataformas usando nada más los agentes móviles que transportan la información de *bundles* o la información de aplicación, es decir los agentes convencionales. Por ello, se han definido un tipo de agente móvil especial cuya función es viajar de plataforma en plataforma manteniendo los criterios actualizados.

Esta estructura común en la que los agentes móviles pueden leer y escribir es una primera fuente de problemas de seguridad. La información está estructurada en forma de árbol, donde cada rama de éste es una ontología distinta, como se puede ver en la figura 4. Agentes móviles pertenecientes a la ontología de la medicina, por ejemplo, debería poder leer o escribir nada más bajo la *rama* perteneciente a la ontología médica. Tanto si se trata de los agentes especiales, como de los convencionales, se necesita garantizar por tanto, que el acceso a la estructura se cumpla la confidencialidad, la autenticación, la integridad y el no repudio.

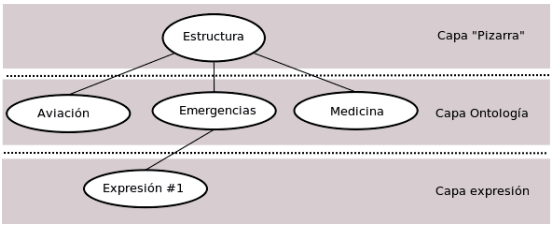


Fig. 4. Organización de la estructura en árbol de ontologías

Los agentes móviles son priorizados en función de sus propiedades. Un agente móvil pertenece a uno o más dominios u ontologías. Sus propiedades se calcularán en función de las diferentes expresiones asociadas a las ontologías. Tenemos dos posibilidades para calcular estas prioridades. Por un lado podríamos dejar que la plataforma local consultase las variables locales de todos los agentes móviles y calcular sus prioridades. Por otro lado, una segunda opción sería dejar que los agentes móviles calculasen sus propiedades ellos mismo e informasen de ella a la plataforma local antes de su migración.

Se ha de tener en cuenta que en ambas opciones los agentes móviles podrían falsificar estas prioridades. En la primera opción, un agente móvil podría llegar a modificar sus variables locales para obtener una mayor prioridad de la que merece. En cambio, en la segunda opción, el agente móvil podría directamente declarar a la plataforma local una prioridad inmerecida.

Las prioridades falsificadas no deberían afectar a otros agentes móviles de otros aplicaciones. Por lo tanto, los agentes móviles pertenecientes a diferentes aplicaciones no deberían competir directamente entre ellos. La propuesta es incluir en el planificador una estructura *round robin* que contenga una cola por aplicación. La plataforma recorre la estructura obteniendo el primero o los n primeros elementos de cada cola, en función del peso asignado a cada aplicación. Esta peso es determinado por la importancia cada aplicación y está controlado por la plataforma local.

En la figura 5 cuatro colas que pertenecen a cuatro aplicaciones diferentes se definen. El planificador permite migrar en cada turno un número diferente de agentes móviles de las diferentes colas, siguiendo la estructura *round robin*.

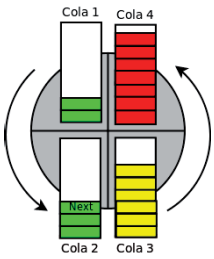


Fig. 5. Estructura *round robin* con las colas de las aplicaciones

B. Congestión DTN

Un problema muy común en todo tipo de redes es el problema de la congestión. Las capas tradicionales de TCP/IP intentan solventar el problema desde varias de sus capas. Por una lado TCP trata los problemas de extremo a extremo e intentar encontrar una velocidad óptima de comunicación, mientras que IP, que se ocupa de los nodos intermedios, maneja colas y políticas de descarte de datagramas cuando estas colas se saturan.

En el caso de redes DTN, la congestión creada por flujos de datos con un ritmo elevado es un problema secundario, por la misma definición de red DTN. En la capa DTN nos enfrentamos a problemas similares a los de IP. Los agentes móviles que transportan *bundles* o bien la información de aplicación, permanecen acumulados en nodos DTN hasta que otro nodo DTN acepte su custodia. Si el *buffer* en el que los agentes móviles se acumulan se llena, tiene que haber un criterio para ver qué agente móvil debería ser descartado. Nuestra propuesta para llevarlo acabo es dinámica, permitimos que los mismos agentes móviles propaguen estos criterios.

En la figura 6, se puede ver como un agente móvil transporta la información sobre la congestión de la plataforma en la que se halla. Esta plataforma se encuentra saturada de agentes móviles tal y como se ve en la representación de su cola. El agente móvil transporta esta información sobre la congestión a su plataforma destino de tal modo que futuros agentes móviles dispongan de esta información y elijan como plataforma destino plataformas alternativas a la plataforma congestionada.

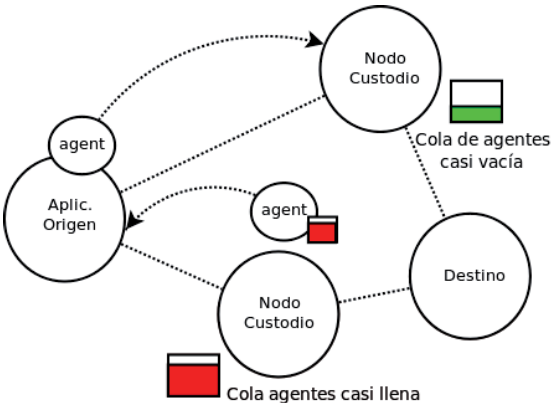


Fig. 6. El agente móvil lleva información sobre la congestión de plataforma en plataforma

La desventaja de este modo de controlar la congestión es que un agente móvil podría informar a una plataforma, de forma fraudulenta, sobre una inexistente congestión en una plataforma vecina. Consecuentemente, esta plataforma vecina sería ignorada por los algoritmos de encaminamiento, lo cual podría dar lugar a una denegación de servicio de esta plataforma o congestión de otras plataformas.

C. Control de flujo

Dentro del estándar del protocolo *Bundle protocol* existe la opción del control del flujo, que es una manera de orientar a niveles inferiores al *bundle* con el objetivo de obtener una determinada calidad de servicio. En el caso de que agentes móviles transporten *bundles*, el objetivo es definir una correspondencia entre la identificación del flujo en el *bundle* y las prioridades de los agentes móviles asociados con los *bundles*

Por un lado tenemos la prioridad definida por el emisor de la aplicación incluida en la cabecera del *bundle*, tal y como se define en [19]. Esta prioridad es estática y no tendría sentido modificarla, pero puede ser tenida en consideración a la hora de calcular la prioridad del agente móvil asociado al *bundle*. Pero si lo que queremos es dar una cierta calidad de servicio a un cierto flujo de *bundles*, podremos usar el *flow label* definido en [19]. En nuestra implementación del planificador se ha incluido una manera de dar prioridades razonablemente parecidas a agentes móviles dentro del mismo flujo.

Un agente móvil podría falsear el flujo al que pertenece con el fin de obtener una determinada calidad de servicio no merecida. Por tanto, se necesita definir mecanismos de seguridad que garanticen la autenticidad del identificador del flujo para evitar posibles falsificaciones de éste.

D. Capacidad de los recursos de computación

Todos los agentes móviles que corren en una plataforma comparten los mismos recursos de cálculo. El código de un agente puede consumir excesivamente los recursos de la plataforma como los ciclos de la cpu, la memoria o el ancho de banda, de tal modo que la plataforma se vea impedida de dar servicio a otros agentes móviles. Establecer límites tanto de memoria como de cpu para los agentes móviles es un tema delicado. Se podría limitar el buen funcionamiento de agentes no fraudulentos.

E. Capacidad de almacenamiento dedicado a los agentes

El tamaño de la colas de almacenamiento de agentes móviles es limitado. En el caso de que se llegue al límite de la cola, nuevos agentes móviles o incluso agentes encolados, tendrán que ser descartados. Esto es un manera muy fácil de provocar una denegación de servicio. Basta inundar una plataforma con agentes móviles, que la plataforma tendrá que descartar agentes móviles. Los agentes encolados descartados serán aquellos con prioridad más baja. Individuar a partir de los agentes móviles, cuáles son parte de la *inundación* es complicado, ya que no todos pueden venir del mismo origen ni puede que tengan ningún patrón en común.

F. Autenticación de las plataformas

Por un lado, necesitamos tener la seguridad que las migraciones de los agentes móviles son secretas y autenticadas, además que se debe asegurar el no repudio de éstas. Por otro lado, se tiene que elegir si las plataformas pueden albergar cualquier agente móvil o debería ser un ambiente cerrado y autenticado. En el caso de las redes convencionales como internet, los routers se incorporan a las redes como un sistema abierto. Creemos que en ambientes DTN debería ser igual. Es evidente que esto comporta problemas de seguridad inevitables como plataformas que aceptan agentes móviles con el fin de provocar denegaciones de servicio o por tener acceso físico a éstas.

IV. Hacia una arquitectura segura en redes DTN basadas en agentes móviles

Existe abundante literatura que propone soluciones para aplicaciones basadas en agentes móviles, como por ejemplo [21] o [24]. Tratan temas generales como seguridad en las plataformas, ejecución segura de agentes móviles, confidencialidad de los datos de los agentes, integridad del código del agente, transporte de llaves, confidencialidad en la comunicación entre agentes, etcétera. Las soluciones propuestas para muchos de los problemas se apoyan en procedimientos de seguridad clásicos, como puede ser sistemas asimétricos PKI basados en llave pública y llave privada.

La falta de conectividad simultánea que se observa de manera generalizada en las redes DTN dificulta el uso de infraestructuras similares a PKI. Por ejemplo, en el caso de la comunicación entre agentes, al tratarse éste de un servicio proveído por las plataformas, una de las soluciones clásicas que se utilizan para garantizar están basadas en una PKI. Cada agente dispone de un certificado expedido por una autoridad certificadora (CA). Esto representaría un problema en caso de usarse sobre una DTN, ya que para poder cifrar la información o comprobar una firma se necesita tener acceso a las llaves públicas de las partes implicadas. Debido a la posible falta de conectividad, en ambientes DTN no se puede asumir el acceso a los servidores que proveen estas llaves, así como tampoco podemos garantizar disponer de listas actualizadas con los certificados revocados por las autoridades certificadoras (listas *CRL's*).

La solución propuesta en [9] a este problema se basa en la criptografía basada en identidades (*Identity based Cryptography*). Las llaves públicas no se obtienen, sino que se generan a partir de cadenas identificadoras conocidas, como el propio nombre del agente o el nombre de la plataforma. Las llaves públicas son generadas por los PKG (*Private Key Generator*). Cada llave es válida por un cierto periodo, un día por ejemplo. Por tanto, con este tipo de sistema criptográfico las listas de revocación no son necesarias, ya que son sustituidas por una actualización de los identificadores. En el caso de los agentes móviles, estos identificadores podrían ser el nombre de la plataforma o el nombre del agente concatenado con el periodo de validez de la llave. Por ejemplo, `agent@platform.uab.cat-7-9-2010`. Ésta será la llave pública con la que cualquier agente móvil podrá cifrar información con destino al agente móvil `agent@platform.uab.cat`, o autenticarlo. Del mismo modo, se puede utilizar para comprobar la validez de firmas en general, con aplicaciones directas a muchos de los problemas de seguridad expuestos en la sección anterior.

Sin embargo, la propia naturaleza de las redes DTN hacen que la IBC tampoco resulte una solución ideal. Los retrasos en la transmisión son tan importantes en algunas DTN que podrían contarse en días o meses. Este escenario complica, por ejemplo, la validación de la autenticidad de unidades de información que fueron firmadas con anterioridad al periodo actual de validez.

Usar agentes móviles para implementar redes DTN tiene la ventaja añadida de poder usar las soluciones criptográficas diseñadas específicamente para ellos. Por ejemplo, la auto-protección de agentes [25], en la que el mecanismo de protección del agente se encuentra en su propio código. De esta manera podría protegerse (privacidad, integridad, autenticidad, etc.) la unidad de transporte de información y la información intercambiada entre nodos sobre encaminamiento, por ejemplo. La auto-protección de agentes permite además, la coexistencia de mecanismos diferentes. Así, diversas aplicaciones con necesidades de seguridad diferentes podrían usar esquemas de protección independientes, sin tener que alterar los nodos de encaminamiento.

Cuando no es posible utilizar ninguna de las soluciones de seguridad disponible, por cuestiones propias de la red o de la aplicación, siempre puede recurrirse a esquemas basados en modelos de confianza [26]. La reputación, por ejemplo, ganada por los nodos en un histórico reciente de interacciones podría usarse de manera combinada con un sistema dinámico de recomendaciones para valorar el riesgo de atentados contra la seguridad de las comunicaciones, o para aplicar técnicas de ostracismo a los nodos problemáticos.

V. Conclusiones y trabajos futuros

En este artículo se ha presentado una arquitectura DTN basada en agentes móviles como alternativa no excluyente a la tradicional propuesta por el protocolo *Bundle protocol*. La aportación fundamental de la nueva arquitectura es la capacidad de los agentes de ejecutar código a la vez que transportan información de aplicación. Esta arquitectura incluye un planificador de agentes móviles con el fin de permitir a agentes considerados importantes poder migrar lo antes posible. Estas propuestas ayudarán a mejorar aspectos de las redes DTN como el encaminamiento, la congestión, el control de flujo, la fiabilidad y a disminuir los retrasos, pero en cambio serán la principal fuente de problemas de seguridad.

El objetivo es garantizar en todas las operaciones derivadas de nuestra arquitectura propuesta, la confidencialidad, la autenticación, la integridad y el no repudio. Mecanismos como los sistemas tradicionales PKI no son válidos por la intermitente falta de conexión.

Como propuestas generales para resolver estos problemas de seguridad se plantean tres mecanismos. El primero fun-

damentado en la criptografía basada en identidades (*Identity based Cryptography*). Añadir el periodo de validez a las cadenas identificadoras conocidas permite prescindir de las listas de revocación, pero conllevan ciertas limitaciones. El tamaño de estas cadenas es extremadamente pequeño si lo comparamos con una llave PKI tradicional. Un ataque por fuerza bruta permitiría en un tiempo razonable obtener su correspondiente llave privada. Incluso si la validez de este par de llaves es de un tiempo determinado, como trabajo futuro se tendría que evaluar cómo gestionar comunicaciones que sean extremadamente perdurables, más allá de este tiempo de validez y sin permitir ataques de fuerza bruta.

Sea cual sea la solución de seguridad escogida para una aplicación sobre DTN cualquiera, la utilización de los agentes móviles como base de la arquitectura de la red añade la ventaja de poder utilizar de manera simultánea varios esquemas. Esto se consigue con el segundo mecanismo propuesto mediante técnicas de auto-protección.

El tercer y último punto a mencionar son los modelos de confianza, que podrían usarse en casos extremos donde los agentes tengan dificultades para obtener información para el uso de esquemas de seguridad (como por ejemplo llaves).

REFERENCES

[1] Scott, K. and S. Burleigh *Bundle Protocol Specification* RFC 5050, November 2007

[2] *Delay Tolerant Networking Research Group* www.dtnrg.org

[3] JE White *Mobile agents make a network an open platform for third-partydevelopers* Computer Nov 1994 Volume: 27, Issue: 11 On pages: 89-90, ISSN: 0018-9162 International Journal of High Performance Computing Applications, Vol. 15, No. 3, 200-222 (2001)

[4] Bellifemine, F., Rimassa, G., Poggi A. *JADE - A FIPA-compliant Agent Framework* Proceedings of the 4th International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents, London, 1999

[5] D. Waitzman. IP over Avian Carriers with Quality of Service. *RFC 2549 (Unrecommended Standard), April 1999.*

[6] S. Dimitriou, V. Tsaoussidis *Effective Buffer and Storage Management in DTN Nodes* http://comnet.ee.duth.gr/comnet/files/edtn09.pdf

[7] Dan Henriksson et. al. *A Caching-Based Approach to Routing in Delay-Tolerant Networks* Proceedings of 16th International Conference on Computer Communications and Networks, (ICCCN) 2007

[8] S. Burleigh et al. *Licklider Transmission Protocol* Work in progress as an IETF internet draft, April 2007

[9] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott y C. Luo. *Applicability of identity-based cryptography for disruption-tolerant networking* In MobiOpp 2007

[10] A Seth, S Keshav *Practical security for disconnected nodes* 1st IEEE ICNP Workshop on Secure Network Protocols

[11] D Jea1, A Somasundara1, M Srivastava *Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks* Lecture Notes in Computer Science, Distributed Computing in Sensor Systems, 2005

[12] M Seligman, K Fall, P Mundur *Storage routing for dtn congestion control* Wireless Communications & Mobile Computing, Volume 7 , Issue 10 (December 2007)

[13] Amir Krifa et. al. *Optimal Buffer Management Policies for Delay Tolerant Networks.* SECON. 2008

[14] Niek J. E. Wijngaards, B. J. Overeinder, M. van Steen *Supporting internet-scale multi-agent systems* Data Knowledge Engineering 41(2-3):229¿245, 2002

[15] A. Lindgren, K.S. Phanse *Evaluation of Queueing Policies and Forwarding Strategies for Routing in Intermittently Connected Networks* First International Conference on Communication System Software and Middleware, 2006. Comsware 2006

[16] C Borrego, S Robles *Relative Information in Grid Information Service and Grid Monitoring Using Mobile Agents* 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009)

[17] J Cucurull *Efficient Mobility and Interoperability of Software Agents* Phdthesis, Universitat Autònoma de Barcelona(2008)

[18] R Martí, S Robles, A Martín-Campillo, J Cucurull *Providing early resource allocation during emergencies: The mobile triage tag* Journal of Network and Computer Applications Volume 32, Issue 6, Pages 1167-1182, November 2009

[19] Network Working Group S. Burleigh *Bundle Protocol Extended Class Of Service (ECOS)* Internet-Draft, December 2010

[20] S. Farrell, V. Cahill. *Delay-and Disruption-Tolerant Networking* Artech House, Inc. Norwood, MA, USA, 2006

[21] W. M. Farmer, J. D. Guttman, and V. Swarup *Security for Mobile Agents: Issues and Requirements* Proc. 19th Nat'l. Info. Sys. Security Conf., Baltimore, MD, Oct., 1996

[22] Psaras, I., Wang, N., Tafazolli, R. *Six years since first DTN papers. Is there a clear target?* In: 1st Extreme Workshop on Communication (ExtremeCom2009), Laponia, Sweden. (2009)

[23] Fulu Li,Nabil Seddigh,Biswajit Nandy, Diego Matute *An Empirical Study of Today's Internet Traffic for Differentiated Services IP QoS* Proceedings of the Fifth IEEE Symposium on Computers and Communications (ISCC 2000)

[24] Garrigues, C., Robles, S., Borrell, J., Navarro-Arribas, G. *Promoting the development of secure mobile agent applications*, Journal of Systems and Software, In Press. Available online 4 November (DOI: 10.1016/j.jss.2009.11.001), (2009).

[25] Ametller, J., Robles, S., Ortega, J. *Self-Protected Mobile Agents*, In 3rd International Conference on Autonomous Agents and Multi Agents Systems. ACM Press, vol. 1, N.R. Jennings, C. Sierra, L. Sonenberg, M. Tambe, 362-367, (2004).

[26] Sabater, J., Sierra, C. *Review on Computational Trust and Reputation Models*, Journal Artificial Intelligence Review. Springer, Issue Volume 24, Number 1, (2005).

Implementación de IPsec en una arquitectura *TCP splitting*

Juan Caubet, Jose L. Muñoz, Juanjo Alins, Jorge Mata-Díaz, Oscar Esparza
Universitat Politècnica de Catalunya (UPC)

***Resumen*—El rendimiento de las aplicaciones que utilizan el protocolo de transporte TCP (*Transmission Control Protocol*) sobre enlaces vía satélite tiene una degradación significativa. Esto se debe principalmente a que el algoritmo de control de congestión estándar de TCP no es adecuado para superar las deficiencias de las redes satelitales. *TCP splitting* es una solución prometedora para mejorar el rendimiento general de TCP, incluso en el segmento satelital. La división de la conexión TCP se logra mediante la instalación de dos PEPs (*Performance Enhancement Proxies*) en los extremos del segmento satelital. Sin embargo, la división de TCP entra en conflicto con IPsec. Si el cifrado y/o la autenticación son aplicados sobre los datagramas IP, el PEP no puede manipular las correspondientes cabeceras IP y TCP para dividir las conexiones TCP. En este trabajo presentamos tres propuestas para implementar IPsec en un escenario *TCP splitting*, proporcionando los servicios de seguridad habituales y un buen rendimiento en la conexión vía satélite. La idea básica es permitir a los PEPs manipular las cabeceras IP y TCP en función del nivel de confianza que los usuarios tengan en ellos.**

I. INTRODUCCIÓN

Las redes de banda ancha vía satélite están ganando importancia debido a su alta disponibilidad de ancho de banda y gran cobertura. Estas redes satelitales jugarán un papel crucial en el futuro de Internet debido a la necesidad de servicios de comunicación en cualquier momento y en cualquier lugar. Sin embargo, se ha demostrado que el protocolo TCP (*Transmission Control Protocol*) tiene una degradación significativa sobre enlaces satelitales. Esto se debe principalmente al hecho de que las redes con enlaces satélite presentan grandes retardos de propagación, introducen una alta probabilidad de error de transmisión y disponen de un notable nivel de asimetría entre los anchos de banda de los canales de difusión y de retorno.

La degradación de TCP se debe principalmente a que su algoritmo de control de congestión no es adecuado para superar las deficiencias de los enlaces satelitales [1], [2]. TCP aumenta su ventana de congestión hasta que se produce una pérdida. Entonces, cuando ésta es detectada, el número de paquetes dentro del sistema se reduce a la mitad. En las redes terrestres, las pérdidas de paquetes son causadas principalmente por la congestión en las colas de espera de los dispositivos de red. No obstante, las pérdidas de paquetes también pueden ser causadas por errores de transmisión. Este efecto es especialmente notable en las redes inalámbricas, provocando una reducción innecesaria de la carga del sistema y, por lo tanto, del rendimiento del protocolo TCP. Esta degradación se ve acentuada cuando la red inalámbrica además dispone de un elevado RTT (*Round Trip Time*). Así, en redes con satélites geoestacionarios, las conexiones TCP pueden tardar varias decenas de segundos en restituir la carga de paquetes tras una pérdida de paquetes debido a errores de transmisión.

Se han propuesto algunas soluciones para superar los problemas de TCP sobre enlaces satelitales [3]. Por un lado, algunas extensiones del TCP estándar han sido propuestas especialmente para las redes vía satélite. Y por otra parte, algunas variantes de TCP han sido especialmente diseñadas para optimizar su rendimiento en entornos satelitales, como *TCP Peach*, *TCP Westwood* o *TCP Hybla*.

Sin embargo, la eficacia de estos dos tipos de soluciones está limitada por el hecho de que no están universalmente adoptadas por todos los sistemas finales en Internet.

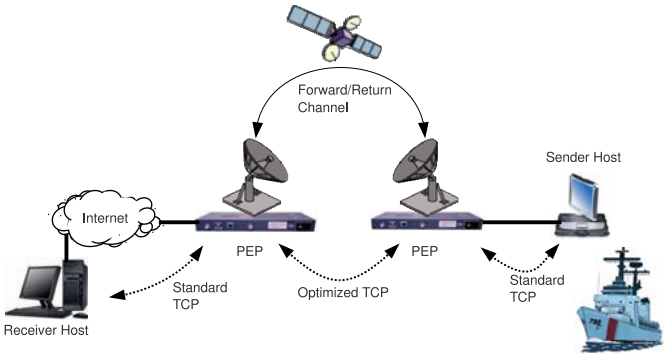


Figura 1. *TCP splitting*.

Los PEPs (*Performance Enhancement Proxies*) pueden ser introducidos para utilizar esos protocolos optimizados, o extensiones, sobre enlaces vía satélite, sin variar los TCPs de los segmentos cableados [4]. El objetivo es dividir la ruta completa en segmentos cableados y un segmento satelital. En general, la comunicación se divide en tres partes o conexiones: emisor-PEP, PEP-PEP y PEP-receptor. Este mecanismo también se conoce como *TCP splitting*. La figura 1 muestra un escenario típico de esta técnica, en la que un barco (que actúa como servidor) proporciona contenidos a un host situado en Internet.

El objetivo de la división es aislar el enlace satelital de gran latencia mediante la introducción de agentes intermedios (PEPs), que dividen la conexión TCP. Los PEPs son responsables de la recepción, el almacenamiento, y el reconocimiento de los datos generados por un emisor, y además, del reenvío de éstos hacia el receptor. *TCP splitting* permite implementar un control de congestión optimizado para mejorar el rendimiento

de TCP en la conexión vía satélite, y dejar el protocolo TCP estándar en los segmentos cableados. Esta división es transparente tanto para el origen como para el destino de la comunicación.

Las redes satelitales también son propensas a ataques de seguridad, debido especialmente a su naturaleza *broadcast*. Por esta razón, es necesario proteger estas comunicaciones. IPsec es una solución estándar que proporciona los servicios de seguridad necesarios para prevenir la mayoría de estos ataques. A diferencia de otras soluciones extremo a extremo que operan en la capa de transporte o en la capa de aplicación, IPsec opera en la capa de red. IPsec se utiliza principalmente para crear VPNs (*Virtual Private Networks*), aunque también se puede utilizar para proteger las comunicaciones entre dos máquinas remotas. En nuestro escenario, el problema es que el uso de IPsec afecta negativamente al funcionamiento de los PEPs, ya que ellos necesitan manipular las cabeceras de los protocolos TCP/IP, que están criptográficamente protegidas. Si se utiliza IPsec, los PEPs no pueden dividir las conexiones TCP y, en consecuencia, el rendimiento de TCP en la redes satelitales no puede ser mejorado.

En este trabajo presentamos tres propuestas diferentes para permitir que IPsec y *TCP splitting* puedan trabajar conjuntamente. El objetivo es proporcionar a las comunicaciones los niveles de seguridad adecuados, sin afectar demasiado las operaciones realizadas por los PEPs. En la primera propuesta, que se explica en la sección III-A, consideramos un escenario particular en el que los usuarios finales (emisor y receptor) no confían en el operador satelital (tal vez porque desconocen su existencia), por lo que quieren que todos los datos intercambiados sean protegidos criptográficamente por IPsec. Así pues, no se confía en absoluto en los PEPs, y por ello no pueden ni leer ni manipular los paquetes TCP/IP, y obviamente, no pueden dividir las conexiones TCP. Sin embargo, estos paquetes son encapsulados por un protocolo TCP optimizado con el fin de mejorar el rendimiento en la conexión satelital. En las otras dos propuestas que se explican en las secciones III-B y III-C, respectivamente, consideramos que los usuarios finales conocen al operador satelital y confían en él. Esto permitirá que los PEPs puedan leer/modificar todo el contenido (*Fully-trusted PEPs*), o ciertas partes (*Partial-trusted PEPs*), de los paquetes TCP/IP, dependiendo del grado de confianza que los usuarios puedan tener en el operador de red y la mejora de rendimiento que quieran conseguir.

II. TRABAJOS RELACIONADOS

Las propuestas que utilizan IPsec en escenarios satelitales con PEPs se pueden clasificar en dos grupos, los que modifican el comportamiento estándar de IPsec y los que sólo se adaptan a él.

En cuanto al primer grupo de propuestas, una de las soluciones más interesantes es ML-IPsec [5], que se basa en dividir los paquetes en diferentes zonas donde se aplicarán los diferentes servicios de seguridad de forma independiente. El número de zonas y su tamaño son definidos a priori utilizando un mapa de zonas. La propuesta también rediseña las Asociaciones de Seguridad (SAs) para definir el tipo de seguridad (algoritmos criptográficos, claves, etc.) que se va a utilizar en cada zona. Se crea una nueva Asociación de Seguridad Compuesta (CSA), que consta de dos partes: una que contiene información común a todas las zonas, y otra que contiene una lista de SAs reducidas, una por zona. Tanto el mapa de zonas como la CSA son compartidas por todos los dispositivos que tienen acceso a alguna zona de los paquetes IPsec. En [6] se puede encontrar un análisis crítico de ML-IPsec. [7] y [8] son dos propuestas que también dividen los paquetes en zonas: una zona para las cabeceras TCP/IP y otra para los datos de usuario. En [8], las dos zonas están cifradas con dos claves diferentes, mientras que en [7] también se utilizan diferentes algoritmos criptográficos.

En referencia al segundo grupo de propuestas, en [9], los dispositivos IPsec establecen una sesión con el PEP para proporcionarle la información necesaria para generar los ACKs prematuros. El PEP utiliza un Identificador de Conexión (CI) para relacionar cada paquete TCP con la información proporcionada por el remitente. Otras soluciones proponen generar un hash de la información de flujo TCP e incluirlo en el campo de opciones de la cabecera IP [10], [11]. En este caso, los PEPs pueden distinguir diferentes flujos de tráfico TCP sin necesidad de modificar los paquetes, por lo que es posible retransmitir los paquetes perdidos en el enlace satelital (*TCP snooping*). En [12], el mecanismo de recuperación de pérdidas de TCP está explícitamente informado sobre la naturaleza de las mismas. En [13], los temporizadores TCP se pueden congelar obligando al emisor TCP a pasar al modo persistente.

III. SOLUCIONES DE SEGURIDAD IPSEC EN ARQUITECTURAS TCP SPLITTING

Como ya se ha comentado, nuestro objetivo es implementar servicios de seguridad con IPsec en arquitecturas *TCP splitting*. El problema es que en estas arquitecturas los PEPs necesitan tener acceso a la información transportada por los paquetes en las cabeceras TCP/IP. Como veremos más adelante, existe un compromiso entre la mejora del rendimiento de la comunicación y la aplicación de la seguridad. También queremos mencionar que no hemos considerado las soluciones que se basan en no aplicar los servicios de seguridad a ciertas partes del paquete o en copiar algunos datos en áreas no protegidas por IPsec. No consideramos este tipo de soluciones porque crean críticas vulnerabilidades de la seguridad. En este caso, las comunicaciones seguras tienen lugar entre entidades de confianza, mientras que los dispositivos intermedios que no son de confianza no tendrán acceso a los datos transportados por los paquetes IPsec.

En este trabajo, proponemos tres maneras de lograr seguridad. La primera propuesta considera que los PEPs no son entidades de confianza. Por otra parte, la segunda y la tercera consideran que los nodos finales tienen cierto grado de confianza en los PEPs. Para estos dos casos, hemos desarrollado un sencillo protocolo de intercambio seguro para que los dispositivos IPsec compartan las Asociaciones de Seguridad (SAs) con los PEPs.

III-A. *Untrusted PEPs*

En esta propuesta los PEPs no se consideran entidades de confianza. Esta consideración hace que no puedan manipular las cabeceras TCP/IP. Esto evitará que los PEPs puedan aplicar los mecanismos correspondientes para mejorar el rendimiento del enlace satelital, como el envío de confirmaciones prematuras a los usuarios finales (*TCP spoofing*). Por esta razón, la única manera de mejorar el rendimiento sin revelar ningún dato sobre el paquete, es encapsularlos en un protocolo TCP optimizado. Con esta solución, los PEPs, al menos, pueden controlar las retransmisiones necesarias sobre el enlace satelital. Más concretamente, este control de retransmisión lo llevan a cabo los PEPs mediante el intercambio de reconocimientos, teniendo la propiedad de gestionar las retransmisiones de paquetes debidas a pérdidas en el enlace satelital de forma transparente a los usuarios finales. Por lo tanto, se evitan los efectos de una reducción innecesaria de la carga en el sistema. Observe que esta propuesta se comportará como un mecanismo típico de *snooping*, pero manteniendo los paquetes IPsec sin cambios.

El hecho de que el PEP deba establecer una conexión TCP optimizada antes de poder enviar información, hace que esta solución requiera del establecimiento de dos conexiones: una conexión TCP estándar entre los usuarios finales y otra conexión TCP optimizada entre los PEPs. El inconveniente es que para establecer conexiones, TCP utiliza un *Three-Way Handshake* que introduce cierto retardo, y por consiguiente, una reducción del tiempo de respuesta de la aplicación de usuario. En la figura 2 se muestra el intercambio de paquetes requeridos por esta propuesta.

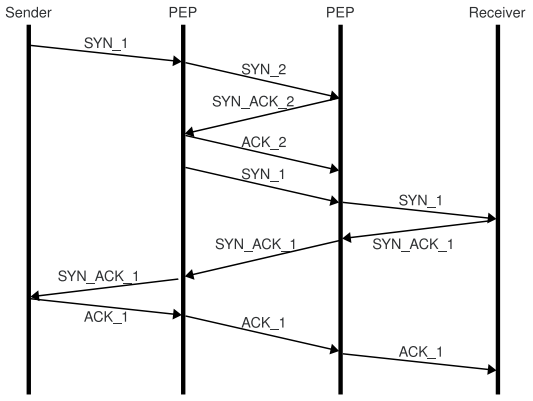


Figura 2. *Three-Way Handshake* en la propuesta *Untrusted PEPs*.

Los paquetes marcados con el "1" son los que se utilizan para establecer una conexión TCP estándar entre los usuarios finales. Por otra parte, los paquetes marcados con el "2" son los que se usan para establecer la conexión TCP optimizada. Un simple análisis permite observar que la fase de establecimiento de la conexión extremo a extremo se incrementa en un RTT del segmento satelital. Este incremento está dentro del margen de operación del protocolo TCP estándar que admite un tiempo de espera del primer paquete SYN de hasta 3 segundos. Una descripción detallada de la operación de los PEPs en esta propuesta sería la siguiente:

1. El primer PEP extrae la cabecera IP de los paquetes que le llegan a la capa de red y los pasa a la capa de transporte.
2. Una vez en la capa de transporte, el PEP añade la cabecera del protocolo TCP optimizado y, a continuación, pasa los paquetes a la capa de red.
3. La capa de red añade la cabecera IP y transmite los paquetes a través del enlace satelital.
4. El otro PEP recibe los paquetes a través del enlace satelital, y una vez en la capa de red, les extrae la cabecera IP y los pasa a la capa de transporte.
5. En la capa de transporte, el PEP genera un paquete ACK para cada paquete recibido.
6. A continuación, el PEP elimina la cabecera del protocolo TCP optimizado y devuelve el paquete a la capa de red.
7. Por último, la capa de red añade la cabecera IP y envía el paquete a su destino final.

Hay que tener en cuenta que los pasos anteriores tienen que ser realizados para cada uno de los dos sentidos de la comunicación, y tanto para paquetes de datos como para paquetes ACK.

Por otra parte, también notar que los paquetes permanecen protegidos desde el inicio de la comunicación hasta el final mediante IPsec. Como resultado, en caso de utilizar el protocolo de seguridad *Authentication Header* (AH) se proporciona integridad de los datos y autenticación, y en caso de utilizar el *Encapsulating Security Payload* (ESP) la confidencialidad está asegurada. Por lo tanto, el nivel de seguridad de esta propuesta es equivalente al proporcionado por el IPsec estándar.

Los dos principales inconvenientes de esta propuesta son:

- El overhead introducido en los paquetes debido al hecho de añadir una nueva cabecera TCP en el enlace satelital.
- Un aumento del retardo debido a la creación y al cierre de las conexiones TCP adicionales entre PEPs.

La propuesta presentada en esta sección puede ser conveniente para escenarios en los que los dispositivos IPsec finales no saben de antemano que sus comunicaciones van a pasar a través de un enlace vía satélite. En tal escenario, los usuarios pueden utilizar IPsec normalmente sin tener que tomar ninguna decisión adicional. Un ejemplo de este tipo de escenario puede ser un *host* ubicado en una red de área local (LAN) que no es consciente de que se está conectando a Internet a través de un enlace satelital.

III-B. *Fully-trusted PEPs*

En esta propuesta se supone que uno de los usuarios finales que utiliza IPsec puede establecer una relación de confianza con uno de los PEPs. Este puede ser el caso de un usuario cuyo ISP (*Internet Service Provider*) es un operador satelital. Aquí suponemos que el usuario puede confiar completamente en los PEPs. Con este tipo de propuesta es posible poner a su disposición la información contenida en las cabeceras.

En este caso, el PEP podrá crear/modificar paquetes IPsec válidos, posibilidad que puede aprovecharse para mejorar el rendimiento de la conexión vía satélite.

Entrando más en detalle, nuestra propuesta funciona de la siguiente forma. En un principio, los usuarios origen y destino establecen una conexión IPsec extremo a extremo mientras los PEP se encuentra en modo pasivo (no están involucrados en esta fase de la comunicación). Una vez que la comunicación IPsec está establecida, y por tanto las SAs se han negociado, uno de los usuarios finales (el que tiene una relación de confianza con los PEP) se las envía a los PEPs[1]. Después de eso, ya se pueden empezar a transferir datos.

En la fase de transferencia de datos de la comunicación, los PEPs operan de la siguiente forma:

1. El primer PEP extrae la cabecera IP de los paquetes que le llegan a la capa de red.
2. Extrae la protección criptográfica de los paquetes antes de enviar los datos de usuario a la capa de transporte.
3. Una vez en la capa de transporte, envía un TCP ACK al usuario origen por cada paquete TCP recibido. Ya que el PEP conoce las SAs, puede crear TCP ACKs válidos para todos los paquetes recibidos (cifrados y/o autenticados). De hecho, el PEP suplanta al destino final.
4. El PEP genera nuevos segmentos usando los datos de usuario extraídos de los paquetes recibidos y los pasa a la capa de red. Sin embargo, estos segmentos son de un TCP optimizado para la conexión vía satélite.
5. En la capa de red les añade la protección criptográfica correspondiente y la cabecera IP. En estos nuevos paquetes IP, la dirección IP de destino no cambia.
6. El PEP transmite estos paquetes IP a través del enlace satelital.
7. El otro PEP recibe los paquetes, les extrae la cabecera IP y la protección criptográfica, y obtiene los datos de usuario para enviarlos a la capa de transporte.
8. Este segundo PEP le envía un TCP ACK al primero por cada paquete recibido.
9. En la capa de transporte, el PEP genera segmentos TCP estándar con los datos de usuario. Estos segmentos se transmiten a la capa de red.
10. La capa de red les añade las protección criptográfica correspondiente y la cabecera IP. En estos nuevos paquetes IP, la dirección IP de destino no cambia.
11. Por último, el PEP transmite los paquetes sobre el enlace terrestre.

La solución propuesta es equivalente al típico *TCP splitting* excepto por el hecho de que tenemos una protección adicional en los paquetes IPsec de los usuarios. Tenga en cuenta que se establecen tres conexiones diferentes: emisor-PEP, PEP-PEP y PEP-receptor. En cada una de estas conexiones, se puede utilizar el TCP más adecuado. Por ejemplo, un TCP estándar para las conexiones terrestres y un TCP optimizado para la conexión vía satélite.

[1]Observe que estas SAs se han de volver a enviar a los PEPs siempre que se hayan renegociado.

La figura 3 muestra el *Three-Way Handshake* que se realiza para establecer estas conexiones. Como se puede observar, el tiempo requerido para establecer la conexión extremo a extremo en esta propuesta es prácticamente el mínimo posible sobre que queda determinado por el retardo de propagación del enlace satelital.

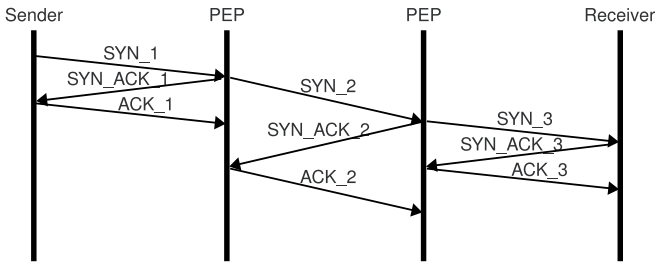


Figura 3. *Three-Way Handshake* en la propuesta *Fully-trusted PEPs*.

En resumen, esta propuesta mejora el rendimiento extremo a extremo porque, por un lado, el PEP puede enviar ACKs prematuros al emisor TCP (evitando reducciones innecesarias de la ventana de congestión), y por otro lado, el PEP puede utilizar un TCP optimizado para la transmisión sobre el enlace satelital. Además, la solución es completamente transparente para el usuario que no es cliente del operador satelital.

Esta propuesta también tiene algunos inconvenientes. Los más importantes son los siguientes:

- Una vez que los PEPs tienen las correspondientes SAs, pueden acceder a toda la información transmitida en los paquetes IPsec (incluyendo los datos de usuario).
- Los PEPs tienen que manipular la protección criptográfica correspondiente (por ejemplo, descifrar cada paquete recibido y cifrarlo antes de transmitirlo).

Los inconvenientes anteriores implican que los PEPs ahora son considerados como terceras partes de confianza (es decir, son nuevos puntos de vulnerabilidad) y que se produce un aumento de la carga de procesamiento en ellos.

III-C. Partially-trusted PEPs (2L-IPsec)

Se propone una tercera propuesta que ofrece un buen equilibrio entre seguridad y rendimiento. Modifica el protocolo IPsec con el fin de proporcionar una protección criptográfica extremo a extremo de los datos de usuario, pero permitiendo que los PEPs puedan manipular las cabeceras TCP/IP para mejorar el rendimiento de la conexión satelital. Se supone que uno de los usuarios de IPsec tienen una relación de confianza con los PEPs. Al igual que en el caso anterior, este puede ser el caso de un usuario cuyo ISP (*Internet Service Provider*) es un operador satelital. Sin embargo, a diferencia del caso anterior, es suficiente una relación de confianza más débil entre los usuarios y los PEPs, ya que el PEP no manipula los datos de usuario, sólo las cabeceras TCP/IP.

Nuestra propuesta se basa parcialmente en las propuestas ML-IPsec [5], LES [7] y [8], ya que también dividen los paquetes en diferentes zonas. Sin embargo, estas partes corresponden a diferentes capas en lugar de a distintas zonas.

La idea es usar capas de cifrado, ya que nos permite divulgar selectivamente diferentes partes de un paquete a los distintos usuarios sin poner en peligro la seguridad de las otras partes. En nuestra propuesta, llamada 2L-IPsec (*two-layer IPsec*), una clave se utiliza para cifrar las cabeceras TCP/IP y otra para cifrar los datos de usuario. Los usuarios finales distribuirán la primera clave para que los PEPs puedan manipular las cabeceras TCP/IP y mantendrán en secreto la segunda, por lo que los PEPs no serán capaces de romper la protección criptográfica de los datos de usuario.

En primer lugar, los usuarios finales de IPsec deben negociar la Asociación de Seguridad (SAs), mediante el protocolo estándar *Internet Key Exchange* (IKEv2) que proporciona IPsec, mientras los PEPs permanecen en modo pasivo. Estas SAs contendrán la clave criptográfica K_D, que se utilizará para proteger los datos de usuario. Una vez que tengan las SAs, tienen que generar una nueva clave de cifrado para proteger las cabeceras TCP/IP, K_H, y enviársela a los PEPs[2].

Entonces, los usuarios finales tienen acceso a los paquetes enteros, y los PEPs sólo pueden manipular las cabeceras TCP/IP y no los datos de usuario. Esto es suficiente para aplicar las técnicas de *splitting*.

En la fase de transferencia de datos de la comunicación, los PEPs operan de la siguiente forma:

1. El primer PEP extrae la cabecera IP de los paquetes que le llegan a la capa de red.
2. Extrae la protección criptográfica de las cabeceras TCP/IP, y envía los datos de usuario (que son protegidos criptográficamente usando K_D) a la capa de transporte.
3. También envía un TCP ACK al usuario origen por cada paquete TCP que recibe. Ya que el PEP conoce K_H, puede crear ACKs criptográficamente protegidos.
4. En la capa de transporte, el PEP genera nuevos segmentos usando los datos de usuario (protegidos), pero utilizando un TCP optimizado para la conexión vía satélite. Estos nuevos segmentos se transmiten a la capa de red.
5. La capa de red les añade la protección criptográfica correspondiente y la cabecera IP (usando K_H). En estos nuevos paquetes IP, la dirección IP de destino no cambia.
6. El PEP transmite los paquetes IP sobre el enlace satélite.
7. El otro PEP recibe los paquetes, extrae la cabecera IP y la protección criptográfica (utilizando K_H), y obtienen los datos de usuario (protegidos) para enviarlos a la capa de transporte.
8. En la capa de transporte, envía un TCP ACK al primer PEP por cada paquete que recibe.
9. Genera segmentos TCP estándar con los datos de usuario (protegidos). Después los transmite a la capa de red.
10. La capa de red les añade la protección criptográfica correspondiente y la cabecera IP, usando K_H. En estos nuevos paquetes IP, la dirección IP de destino no cambia.
11. Por último, el PEP transmite los paquetes sobre el enlace terrestre.

[2]Igual que en el caso anterior, esta operación se debe repetir cada vez que las SAs sean renegociadas.

Observe que en el caso de que un paquete se pierda en cualquiera de las tres partes de la ruta, el paquete sólo se retransmitirá en dicha parte. Por otro lado, la definición de las zonas ha sido específicamente diseñada teniendo en cuenta que para aplicar *TCP splitting* el PEP sólo necesita acceso a las cabeceras TCP/IP. Por otra parte, el tamaño de estas cabeceras es variable, ya que pueden incluir información opcional. Así, hemos definido una zona que contiene las cabeceras TCP/IP y otra zona que cubre el *payload* de TCP, ambas de longitud variable. De esta manera, los PEPs siempre tienen acceso a la zona deseada sin importar el tamaño de las cabeceras, ya que su tamaño se gestiona de forma dinámica.

La figura 4 muestra el formato de las cabeceras IPsec (AH y ESP) en los paquetes de 2L-IPsec (en modo de transporte). La longitud de las cabeceras, el *offset* a la siguiente cabecera y el tipo de protocolo de transporte encapsulado en el datagrama IP se determinan mediante cálculos matemáticos simples.

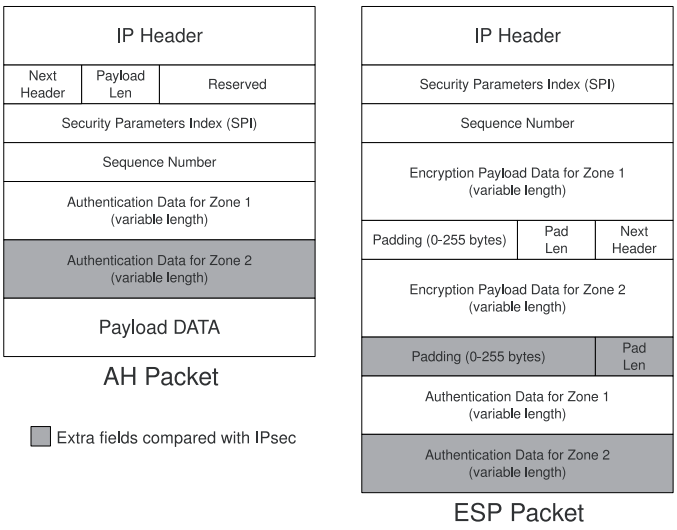


Figura 4. Formato de los paquetes 2L-IPsec (en modo transporte).

Algunas de las ventajas de utilizar 2L-IPsec se resumen a continuación:

- *No compromete la seguridad de los datos de usuario*. Esta propuesta preserva la seguridad de los datos de usuario entre los dispositivos 2L-IPsec. También proporciona confidencialidad, integridad y autenticidad del origen de los datos.
- *Seguridad reforzada de los paquetes*. Ahora se tienen que romper dos claves diferentes para acceder a todos los datos contenidos en el paquete.
- *Preservar la confidencialidad extremo a extremo*. Por ejemplo, si un PEP no necesita modificar ningún dato de los paquetes, entonces las SAs únicamente tendrán que incluir la clave de autenticación, pero no la clave de cifrado, para la zona correspondiente.
- *Son posibles las comunicaciones seguras utilizando TCP splitting*. Este es uno de nuestros principales objetivos.

Tenga en cuenta que con esta propuesta podemos implementar un mecanismo que controle qué partes específicas del paquete pueden ser leídas, modificadas, etc. por terceras partes involucradas en la comunicación. Además, nos permite generar los paquetes ACK prematuros requeridos por *TCP splitting*. Por último, también se pueden retransmitir los paquetes perdidos de manera independiente en cada parte de la ruta.

Como se indica en [7], se ha demostrado que el rendimiento del cifrado de seguridad por capas es comparable al de IPsec.

Nuestra propuesta también tiene algunas desventajas que deben ser consideradas, así:

- *2L-IPsec no es una solución estándar*. La mayoría de los dispositivos conectados a Internet no tendrán este nuevo protocolo. Esto limitará su uso a determinados escenarios, en los que es posible su implantación. Por ejemplo, un escenario real podría ser la comunicación de dos sedes de la misma empresa que utiliza un enlace vía satélite para sus comunicaciones. En este caso, los *routers* de las sedes son los usuarios IPsec finales. Estos dispositivos son los que podrían tener instalado el protocolo 2L-IPsec para mejorar el rendimiento de dichas comunicaciones.
- *Overhead*. Nuestra propuesta aumenta el tamaño de los paquetes, cosa que afecta negativamente al rendimiento. Además, 2L-IPsec requiere más operaciones de cifrado/descifrado, así que son necesarias más CPU y más memoria. Sin embargo, añadir seguridad siempre significa que hay que añadir algo de sobrecarga al sistema.
- *Complejidad*. Los usuarios finales deben descifrar y cifrar los paquetes en dos zonas diferentes utilizando dos claves en vez de una.
- *Distribución de las claves*. Las claves tienen que distribuirse a los PEPs de forma segura.
- *Generación de nuevas claves*. Los usuarios finales tienen que generar claves adicionales. Una solución bastante simple podría ser el uso de funciones hash sobre la clave negociada, H_D, es decir, $K_H = h(K_D)$.

IV. CONCLUSIONES

En este trabajo hemos analizado y propuesto soluciones para utilizar dos mecanismos que al ser aplicados conjuntamente provocan un conflicto. Estos mecanismos son la seguridad de red extremo a extremo proporcionada por IPsec y la optimización del rendimiento de las redes satelitales mediante *TCP splitting*. Por una parte la seguridad extremo a extremo normalmente utiliza criptografía en la capa de red para proteger los datagramas de usuario, y por otra *TCP splitting* requiere que los nodos intermedios puedan realizar operaciones "inteligentes" sobre los paquetes TCP para mejorar el rendimiento. El problema es que algunas partes de los paquetes que son necesarias para lograr las mejoras de rendimiento podrían no ser accesibles debido a la protección aplicada por IPsec.

En general, la situación anterior es un problema difícil de tratar y hay que asumir que no existe una solución adecuada para todos los escenarios. En este sentido, lo mejor que podemos hacer es encontrar un conjunto de soluciones que ofrezcan diferentes compromisos entre seguridad y rendimiento.

En este trabajo, hemos analizado las que consideramos las tres principales propuestas para solucionar el problema. Estos planteamientos conducen a tres ventajas y desventajas diferentes, que se han discutido a lo largo del trabajo. Por último, las diferentes propuestas también se pueden relacionar con varios escenarios posibles, también descritos en el documento.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Educación y Ciencia gracias a los proyectos CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", TSI2007-65393-C02-02 "ITACA" y TEC2008-06663-C03-01 "P2PSEC", y por la Generalitat de Catalunya gracias al grupo de investigación consolidado 2009 SGR 1362.

REFERENCIAS

[1] T. R. Henderson and R. H. Katz. Transport Protocols for Internet-Compatible Satellite Networks. *IEEE Journal on Selected Areas in Communications*, 17(2):326–344, 1999.

[2] C. Caini, R. Firrincieli, M. Marchese, T. de Cola, N. Celandroni M. Luglio, C. Roseti, and F. Potortì. Transport Layer Protocols and Architectures for Satellite Networks. *International Journal of Satellite Communications and Networking*, 25:1–26, 2007.

[3] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, and R.H. Katz. A comparison of mechanisms for improving tcp performance over wireless links. *Computer Communication*, 26(4):256–269, 1996.

[4] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. RFC 3135 (Informational), June 2001.

[5] Yongguang Zhang. A multilayer ip security protocol for tcp performance enhancement in wireless networks. *Selected Areas in Communications, IEEE Journal on*, 22(4):767–776, may 2004.

[6] J. Sing and B. Soh. A critical analysis of multilayer ip security protocol. *Proceedings. Third International Conference on Information Technology and Applications*, 2:683–8, 2005.

[7] M. Karir and J. Baras. Les: Layered encryption security. *Proceedings of the Third International Conference on Networking (ICN'04)*, 2004.

[8] A. Roy-Chowdhury and J.S. Baras. Performance-aware security of unicast communication in hybrid satellite networks. *ICC 2009 - 2009 IEEE International Conference on Communications*, pages 6 pp. –, 2009.

[9] N. Thanthry, M. Deshpande, and R. Pendse. A novel mechanism for improving performance and security of tcp flows over satellite links. *Proceedings - International Carnahan Conference on Security Technology*, pages 197–202, 2006.

[10] D.D. Isci, F. Alagoz, and M.U. Caglayan. Ipsec over satellite links: a new flow identification method. *Proceedings of ISCN'06 7th International Symposium on Computer Networks (IEEE Cat. No.06EX1429)*, pages 140–5, 2006.

[11] A. Parichehreh and B. Eliasi. Vpn over satellite: performance improving of e2e secured tcp flows. *2008 5th IFIP International Conference on Wireless and Optical Communications Networks (WOCN '08)*, pages 40–3, 2008.

[12] V. Obanaik, L. Jacob, and A.L. Ananda. Secure performance enhancing proxy: to ensure end-to-end security and enhance tcp performance over ipv6 wireless networks. *Computer Networks*, 50(13):2225–38, 2006.

[13] G. Ciccarese, M. De Blas, L. Patrono, P. Marra, and G. Tomasicchio. An ipsec-aware tcp pep for integrated mobile satellite networks. *2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No.04TH8754)*, 4:2362–6, 2004.

Esteganografía lingüística en redes sociales. Perspectiva de futuro en lengua española

Alfonso Muñoz Muñoz
Escuela de I.T de Telecomunicación
Universidad Politécnica de Madrid
Email: amunoz@diatel.upm.es

Justo Carracedo Gallardo
Escuela de I.T de Telecomunicación
Universidad Politécnica de Madrid
Email: carracedo@diatel.upm.es

Jorge Ramió Aguirre
Escuela Universitaria de Informática
Universidad Politécnica de Madrid
Email: jramio@eui.upm.es

***Abstract*—El presente artículo analiza las posibilidades de la distribución de estegotextos en redes sociales mediante los últimos avances desarrollados en esteganografía lingüística en lengua española. Existen muchas limitaciones a considerar si se desea ocultar información en textos en lenguaje natural de manera no trivial. En la práctica, la decisión de un procedimiento de ocultación u otro condicionará, desde un punto de vista práctico, el canal (en este caso red social) donde será más fácil transmitir una información oculta; esta característica podría facilitar el trabajo a un potencial estegoanalista. A modo de ejemplo, se analizan algunas características de la red social Twitter.**

I. CONCEPTOS PREVIOS. ESTEGANOGRAFÍA LINGÜÍSTICA

La ciencia de la esteganografía puede ser definida como la ciencia y el arte de ocultar una información dentro de otra que haría la función de tapadera [1], con la intención que la existencia de dicha información no sea percibida. En teoría, sólo quienes conozcan cierta información acerca de esa ocultación (un secreto) estarían en condiciones de descubrirla. Cuando la cubierta es un texto en lenguaje natural ello implica un tipo específico de esteganografía, esteganografía lingüística, y el texto que oculta dicha información es llamado estegotexto.

La idea de ocultar información en textos en lenguaje natural no es ni mucho menos nueva. En los últimos siglos diferentes procedimientos clasificables en *open codes* (cues, null ciphers, jargon code y grilles) y *semagrams* han sido documentados [2]. Algunos ejemplos famosos son *newspaper code* en la época victoriana o la verja de cardano en el siglo XVI [3].

En la actualidad, la esteganografía lingüística intenta aunar los principios de la ciencia de la esteganografía y la lingüística computacional (análisis automático del contenido textual, el análisis morfosintáctico, generación textual, la lexicografía computacional, descripciones ontológicas, etc.) para crear procedimientos no triviales no basados en la oscuridad. En ese esfuerzo dos amplias líneas de investigación son abordadas [4]: a) la modificación de textos existentes y b) la generación automática de estegotextos.

II. ESTEGANOGRAFÍA EN REDES SOCIALES. ANTECEDENTES

En 2008 [5] generalizando el uso de la aplicación de la herramienta de estegoanálisis StegSecret [6] a Internet, se analizó el potencial de las características de las redes sociales con fines esteganográficos. En este trabajo [5] se analizó en primer caso los procedimientos esteganográficos más útiles para los estegomedios más probables en estas redes, como son el contenido multimedia (imagen, audio, video) y las tecnologías web (html, xml, http). Del mismo modo, se analizó la dificultad por parte de un atacante de acceder a posibles datos esteganografiados si éstos aprovechaban al máximo los mecanismos antibot/captchas y los grupos cerrados de usuario para evitar herramientas automatizadas de detección, así como se destacó la utilidad de distribuir contenido de forma multiportador y multiproveedor para evitar análisis por parte de proveedores con otros fines que los redactados en sus políticas de privacidad.

Es en este contexto donde se decidió profundizar en el potencial de la ocultación de información en lenguaje natural de manera no trivial. La información textual está en todas partes, de forma masiva y con características lingüísticas muy diversas, esto la convierte en lo suficientemente interesante de manera individual y en su distribución en las nuevas redes sociales.

III. POTENCIAL DE LA ESTEGANOGRAFÍA LINGÜISTICA EN ESPAÑOL

En los últimos 5 años las publicaciones sobre esteganografía lingüística en lenguas tan diversas como el inglés, ruso o mandarín van en aumento, con utilidad en la ocultación de información en texto en lenguaje natural y con utilidad en el marcado digital de textos. Por desgracia, las publicaciones en lengua española son escasas. Debido a su potencial, en el último año y medio hemos realizado un esfuerzo en acotar la utilidad de diferentes procedimientos con aplicación en esteganografía lingüística en español.

La utilización del lenguaje natural con fines esteganográficos de manera segura, considerando criterios lingüísticos (léxicos, sintácticos, semánticos, de cohesión, de coherencia, etc.) y esteganográfico-estadísticos (análisis de entropía, análisis de frecuencia de caracteres-palabras, ataques basados en conocimiento de cubierta original y cubierta modificada, etc), no es nada sencillo.

La modificación de textos a menudo suele introducir errores detectables por un lector humano. Este inconveniente inicial

se convierte de utilidad para el productor de estos estegotextos ya que permite cuantificar en una primera instancia cómo de bueno es un procedimiento esteganográfico concreto de manera sencilla. La detección de anomalías por parte de un lector humano puede que sea mayor a las detectadas por algoritmos clasificadores automáticos.

En la práctica, el lenguaje natural, como estegomedio, es muy poco redundante (ruidoso) en comparación con otros estegomedios como son las imágenes o los videos, lo cual hace más complicado la creación de algoritmos robustos de ocultación de información en lenguaje natural (información textual). Esta condición hace que las técnicas de ocultación requieran una gran cantidad de información textual para ocultar una cantidad de información no muy abultada. Este hecho puede hacer que estas técnicas sólo sean interesantes en escenarios limitados. Por ejemplo, en Internet ciertos canales serían más aptos para ocultar información que otros.

En el objetivo de analizar qué técnicas esteganográficas en lenguaje natural para lengua española son más interesantes de detectar se analizaron los siguientes procedimientos englobados en las siguientes dos grandes líneas de investigación.

a) Generación automática de estegotextos.

Una buena alternativa para la ocultación de información en textos en lenguaje natural en español sería la generación automática de estegotextos. Su ventaja fundamental reside en la posibilidad de crear estegotextos únicos para cada comunicación, de forma que se dificulte ataques estadísticos y ataques basados en comparaciones texto original-estegotexto, así como tener un mayor control en la modelización estadística del estegotexto creado.

En la última década los esfuerzos en este sentido se centran, principalmente, en la generación de estegotextos que imiten la gramática (sintaxis) y la estadística de un texto típico en una lengua concreta. Aunque, entre las opciones interesantes detectadas, se pueden establecer propuestas de generación automática de estegotextos basadas en construcciones CFGs (Context-Free-Grammar), derivadas de los principios de la teoría de la gramática generativa propuesta por el lingüística A.Noam Chomsky en la década de los 60, en primera instancia no se abordan estos estudios dado que tienen numerosas limitaciones a sortear, como se analizó en [7], por ejemplo, el problema de la privacidad/compartición/generación de la gramática utilizada, ataques basados en estudio de terminales (información última de cada regla), limitación del contexto de utilización y léxico utilizado, etc.

En su lugar es más productivo en primera aproximación el uso de propuestas basadas en modelos estadísticos para minimizar en mejor medida ataques estadísticos a propuestas de esteganografía lingüística.

Con estas características es interesante el uso de un modelo estadístico N-GRAM con fines esteganográficos. Es decir, dado uno o más textos de entrenamiento es posible anotar las co-ocurrencias de las palabras presentes y medir las frecuencias de repetición de cada una (modelo N-GRAM), es decir, se anota qué palabra viene detrás de otra y con qué probabilidad. El algoritmo de generación de estegotextos elige en cada ocasión una de las palabras disponibles en función de la información a ocultar. Si la selección es aleatoria es más probable que salgan las palabras más probables que las que son menos, y por tanto se imite la estadística de la fuente de entrenamiento. Si se imita la estadística de la fuente es más probable que el estegotexto resultante que está basado en los textos de entrenamiento, tenga validez léxica y sintáctica (validez léxica y sintáctica que deben tener los textos de entrenamiento). Imitar el orden de las palabras tiene utilidad esteganográfica y como demostró Greenberg [8] el orden de las palabras determina hasta tal punto la hechura de una lengua, tanto que no admite trivialidades en su manipulación.

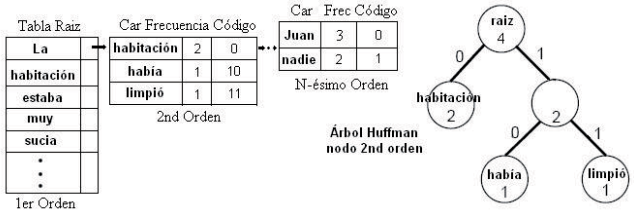


Fig. 1. Algoritmo de generación automática de estegotextos basado en modelo N-GRAM

Esta posibilidad se comprobó en lengua española mediante la implementación de la herramienta Stelin [7]. Aunque depende del texto de entrenamiento, para órdenes de n mayor de 7 y textos de referencia mayores de decenas de KB se obtiene textos con validez léxica y gramatical. El tamaño final de estegotexto generado dependerá de la información a ocultar, de los textos de entrenamiento y del orden n. La capacidad mínima de información podría aproximarse entre 0,5 a 1 bit de información oculta por palabra generada. No obstante, este tipo de algoritmos todavía podrían generar estegotextos con errores gramaticales (depende en mayor medida del texto de entrenamiento). Estos errores podrían minimizarse basándose en los principios de la ley de Zipf (pocas palabras se repiten muchas veces) y en la idea que es posible añadir palabras a medida entre las palabras añadidas en la generación automática de estegotextos sin la necesidad que el receptor las conozca de manera adicional [9]. Por ejemplo, en la Fig1, si el estegotexto creado fuera *La habitación* sería posible añadir palabras entre medias no incluidas en el nivel de la palabra *habitación* (añadir una o más palabras diferentes a *había* y *limpió*). De nuestra experiencia se observa que es posible conseguir estegotextos que oculten menos de 1.000 bits para una relación tiempo-esfuerzo razonable (si se usa edición manual) con una calidad lingüística elevada, que podría dificultar incluso la detección a lectores humanos. Este procedimiento sería de enorme utilidad para la distribución de claves criptográficas, urls, mensajes breves, coordenadas gps, etc. El tamaño del texto resultante dependería de la capacidad editora del emisor, del orden n y de los textos de entrenamiento.

Siguiendo esta idea todavía sería posible dotar al emisor

de mayor libertad en la creación de estegotextos, creando automáticamente estegotextos de peor calidad pero con mayor facilidad de edición (herramienta ryfle) [10]. Un ejemplo de ello consistiría en dividir un texto fuente en grupos de palabras de tamaño W y basar la ocultación en la selección de una de cada grupo (por ejemplo asignamos el mismo peso a cada palabra). Así por ejemplo si la información a ocultar (en bits) fuera I, el número de palabras generadas sería Nmin y el tamaño mínimo necesario (en palabras) de la fuente de entrenamiento sería Source$_{min}$:

$$\mathrm{N}_{min} = I/log_2 W$$
$$\mathrm{Source}_{min} = N_{min} * 2log_2 W$$

Los estegotextos generados no tendrían validez (ni siquiera gramatical), pero para tamaños de grupo (W=4, 8 ó 16) [10] sería fácil añadir palabras antes de cada palabra seleccionada y que no estuvieran contenidas en cada grupo. Esta sencilla idea, el algoritmo es público, proporcionaría una alta calidad lingüística dificultando la tarea de detección a un lector humano (y por tanto a un algoritmo automatizado), si bien a un mayor coste de tiempo-esfuerzo por parte del emisor. Hoy día, se analizan variantes basadas en categorías lingüísticas y co-ocurrencias para facilitar y disminuir el tiempo necesitado en la edición manual [10].

Un procedimiento de este tipo sería ideal para la distribución de una manera altamente segura de claves criptográficas o urls (128-256 bits). Aunque el tamaño del estegotexto resultante dependería de la capacidad editora del emisor, este valor podría aproximarse a 3 ó 4 veces más que el tamaño Nmin, es decir, lo que supondría añadir de 3 a 4 palabras más por palabra generada (con 3 palabras se puede construir en español una oración básica: nombre+verbo+complemento). En [10] se adjunta un ejemplo de estegotexto de 281 palabras, 0,44bits/palabra, que oculta 126 bits de información oculta.

b) Modificación de textos existentes.

La modificación de textos con utilidad esteganográfica con ciertos criterios de seguridad limita su actuación a una serie de procedimientos de ocultación relacionado con el léxico y la semántica, la gramática y la co-ocurrencia (orden de las palabras). Algunas de las propuestas más interesantes que se han documentado son [4]: sustituciones léxico-semánticas, sustituciones sintácticas, aprovechar el ruido de traducir un texto a diferentes idiomas, ocultación basada en formato y errores tipográficos/ortográficos (abreviaturas, acrónimos y otros). En este contexto se analizó las técnicas que parecen ser más productivas en otros idiomas, como por ejemplo los procedimientos en lengua inglesa.

En primer lugar es interesante analizar el potencial de utilizar el orden de las palabras dentro de una frase con utilidad esteganográfica. En [11] se analizó, investigación que sigue en curso, el potencial de las modificaciones sintácticas en lengua española con utilidad esteganográfica y utilidad en natural language watermarking. Basándonos en mediciones en el corpus LEXESP [11] se demuestra cómo en español estructuras, con utilidad en otras lenguas como el inglés, como es la transformación activa-pasiva no tiene utilidad esteganográfica, al no ser la pasiva una estructura lingüística frecuente en español. Este trabajo [11] analiza además la posibilidad de mover ciertos elementos etiquetados previamente (Part of Speech) en una oración. De esta forma a nivel de frase, los resultados indican mayor utilidad esteganográfica en el movimiento de adverbios, especialmente si éste se sitúa al principio o final de frase, y el movimiento de adjetivos con respecto al nombre que modifica (anteponiéndolo o posponiéndolo), más concretamente 1.537 parejas (nombre+adj o adj+nombre) muestran la posibilidad de mover el adjetivo delante o detrás de un nombre y por tanto de ocultar un bit por pareja.

Aunque de momento estas investigaciones están en curso, en media (con suerte) 1 ó 2 bits por frase podrían ser ocultados mediante estos procedimientos. Si bien se podrían generar estegotextos indistinguibles de textos originales, es cierto que sería necesario una cantidad razonable de texto para ocultar información, por ejemplo, 128 frases para 128 bits (entiéndase que un texto podría ser fragmentado en frases considerando puntos, partículas que subordinen oraciones, etc.).

Por otro lado, la ocultación basada en modificaciones léxico-semánticas sería posible. En [12] se analizó la posibilidad de ocultar información mediante la implementación de una herramienta avanzada de sustitución de sinónimos (23.918 sinónimos únicos y soporte para variantes verbales, de género y número). Considerando procedimientos estadísticos para disminuir el impacto de una sustitución de una palabra en el contexto cercano de palabras vecinas se pueden realizar diferentes aproximaciones, basadas en reglas de Word Sense Disambiguation. Entre las propuestas documentadas en [12] una puede basarse en la construcción de tablas estadísticas que alimenten una función de ponderación que cuantifique cómo de bueno es un sinónimo para sustituirlo en un contexto dado. Siguiendo esta idea se podría conseguir una capacidad de ocultación próxima al 29,71% de las palabras totales de un texto. La ocultación mínima en esta situación considerando 1 bit por palabra útil, sería en media de 0,2971 bits/palabra. No obstante, aunque se construya un tabla estadística significativa, nuestros experimentos están basados en un corpus de 120 millones de palabras, y sólo se realizarán modificaciones considerando los dos sinónimos más probables de cada palabra a sustituir, en función de su contexto, todavía sería posible implementar un ataque haciendo uso de un clasificador SVM que detectara al menos el 50% de los documentos con información oculta, documentos de 269 palabras y 52,45 bits/documento.

IV. ESTEGANOGRAFÍA LINGUISTICA EN REDES SOCIALES. MITOS Y REALIDADES

La distribución de información oculta en textos en lenguaje natural mediante procedimientos no basados en oscuridad hace que las opciones disponibles para ocultar información se vean seriamente reducidas. En la práctica estos procedimientos estarán basados en sustituciones léxico-semánticas, uso del orden de las palabras (modelos estadísticos) y el posible uso

de reglas gramaticales. Desde un punto de vista aproximativo y puramente conceptual procedimientos de ocultación de este tipo daría una capacidad mínima de ocultación del orden expresado en la siguiente figura.

Técnica	Capacidad Mínima	Procedimiento Automático
Modelo N-Gram (sin edición manual)	0,5 - 1 bit / palabra	SI
Herramienta Ryfle	0,44 bits / palabra	NO
Sustitución de palabras por sinónimos (uso tablas estadísticas)	0,2971 bits / palabra	SI
Modificaciones sintácticas basadas en el movimiento de palabras	1-2 bits / frase	SI

Fig. 2. Capacidad mínima de ocultación por palabra del estegotexto creado o modificado.

Según estos valores, e independientemente de la seguridad de cada técnica concreta, para ocultar, por ejemplo, una mínima información útil, que podría ser de unos 128 bits, se necesitaría fragmentos de textos de al menos: 128-256 palabras (N-Gram), 290 palabras (ryfle) y 430 palabras (sinónimos). Esta capacidad de ocultación implica la necesidad de textos de cierto tamaño. Por ejemplo, en su aproximación a las redes sociales todas estas técnicas son ideales en la publicación de estegotextos en blogs, más cuestionable puede ser (sin ser troceados o distribuidos) en comentarios en foros, mensajes en redes tipo facebook, tuenti y similares.

Obtener los tamaños medios de los mensajes en este tipo de redes no es sencillo, al estar los usuarios en grupos cerrados, ya que el acceso implicaría anular sistemas de protección o autenticación. En cualquier caso y de forma aproximativa, el tamaño de los comentarios en facebook o tuenti se encuentra entre las 10 y las 25 palabras, excepciones puntuales (en los casos manualmente comprobados) excederían hasta no más de 100 palabras.

En el caso que se utilicen otros procedimientos, basados en técnicas sin documentar/publicar, la modelación del lenguaje, el uso de la estadística y el uso de expresiones lingüísticas podría facilitar la automatización del estegoanálisis de estas técnicas no documentadas. En este sentido, dado que existe un número importante de redes sociales donde el tamaño de la información textual no permitiría (de forma simple) las técnicas documentadas (para ocultar al menos cientos de bits) podría ser interesante explotar nuevas características de ocultación. Por ejemplo, son interesantes las redes sociales basadas en comunicaciones rápidas, frases cortas, con poca preocupación en la calidad lingüística, produciéndose faltas ortográficas y tipográficas en cuantía. Un entorno donde podría ser interesante el uso de procedimientos esteganográficos basados en errores ortográficos y tipográficos sería precisamente este tipo de redes sociales donde el lenguaje no está muy cuidado. Dado que la presencia de faltas de ortografía podría hacer a un lector humano delatar rápidamente anomalías, sobre todo si un texto presenta estadísticamente más faltas de las esperadas incluso en un canal ruidoso, el enfoque se debe centrar, si esta propuesta es considerada en serio, en distribuir la información dificultando la tarea de programas automáticos en clasificar textos originales y textos con información oculta. Aunque no es sencillo articular una propuesta pública basada en estos principios, en [13] se presenta una aproximación interesante y razonablemente seria. La pregunta clave a resolver es ¿cómo sabe un programa automático que existe una falta de ortografía en un texto?. Con esta idea en mente se pueden formular diferentes sistemas de ocultación:

a) Sistemas que usan malformaciones para producir palabras que no existen en un diccionario. El ataque a estas propuestas se simplifica utilizando diccionarios grandes y modelado de estadística de uso de las palabras.

b) Sistemas basados en errores tipográficos y uso de acrónimos, abreviaturas y similares. La detección de estos procedimientos queda supedita a algoritmos similares a los utilizados en la corrección de textos (por ejemplo, en productos ofimáticos). Algunas de las técnicas que se emplean en su detección son medidas basadas en la mínima distancia de edición, modelos estadísticos n-gram, etc.

c) Malformaciones que provocan que una palabra válida se convierta en otra palabra válida (vaca/baca, toro/loro, etc), separar palabras compuestas (saca puntas/sacapuntas), errores gramaticales, etc. La detección de estos procedimientos requiere de técnicas avanzadas de reconocimiento lingüístico, etiquetadores precisos, técnicas de desambigüación y comprensión semántica.

Aparte de la aportación conceptual de [13], razonan una capacidad de ocultación de 1bit/1 palabra útil, no queda lo suficientemente claro el interés práctico de este tipo de propuestas. No ha sido posible avanzar en el análisis de esta propuesta concreta al no poder acceder a la herramienta comentada en [13] y analizar la seguridad real de los estegotextos producidos respecto de un modelo de lenguaje obtenido de textos de entrenamiento, así como ver como se comportarían estos mensajes ocultos respecto de la estadística de un canal concreto donde se transmitieran.

Analizadas muchas de las variantes esteganográficas más razonables en textos en lengua española se va a analizar a continuación algunas características de una red social concreta y se va a razonar si es productiva en términos de esteganografía lingüística.

V. ESTEGANOGRAFÍA LINGUISTICA EN TWITTER

La red social twitter es un buen ejemplo de red social donde el tamaño de los mensajes dificultaría ocultar centenas de bits (sin trocear o distribuir estegotextos) mediante una propuesta esteganográfica pública. En el pasado esta red ha sido utilizada de manera muy diversa, en el envío de información cifrada, la administración de botnets, etc [14][15].

Es en este entorno donde podría ser interesante analizar si también es útil en esteganografía lingüística, conclusiones que

se podrían derivar a otras redes similares.

Si se piensa por un momento en este tipo de redes, es factible la implementación de un número elevado de procedimientos esteganográficos basados en oscuridad y que en muchos casos recuerdan a procedimientos utilizados siglos atrás (utilizar ciertas letras, usar el número de palabras para codificar una información, etc). Por ejemplo, sería posible utilizar procedimientos clásicos de alterar las letras de cada palabra poniéndola en mayúscula o minúscula en función de si se codifica un bit 0 o un bit 1, de esta forma en el mejor de los casos se podría ocultar 140 bits (140 caracteres) por mensaje twitter. Como puede observarse sería fácilmente detectable, no obstante siempre podría ser utilizado como mecanismo para transmitir más información en los mensajes sin fines esteganográficos (en el mejor de los casos 140 bits/7 bits-carácter = 20 caracteres más) teniendo el receptor la capacidad todavía de comprender el mensaje publicado. Sin ser puramente lingüística otras técnicas de ocultación podrían ser factibles. Por ejemplo, usar los sistemas de codificación de urls compactadas utilizadas en Twitter, como por ejemplo el sistema web bit.ly, con fines esteganográficos. Por ejemplo, un enlace compactado del tipo bit.ly/aWvi3x podría ocultar información jugando con la codificación de los mismos (minúscula, mayúscula y números). Si se toma la molestia de que el enlace realmente exista (existen un gran número de ellos), un atacante no sólo debería comprobar que ese enlace apunta en realidad a una página web sino tener la capacidad a) de analizar la semántica de la frase donde está presente y correlarla con el contenido del enlace para ver si el enlace corresponde a lo que se indica o b) establecer algún tipo de criterio estadístico que permita deducir que de la presencia de muchos enlaces de este tipo se puede destacar la presencia de un sistema de codificación concreto. Como se describirá posteriormente la presencia de direcciones web en un mensaje Twitter se produce en un 23,23% de los casos.

En cualquier caso, para analizar mejor el potencial de la esteganografía ligüística en esta red, se decide la construcción de un corpus basado en 103 usuarios españoles de la red twitter, con la condición que tengan al menos 1.000 seguidores y como poco 3.200 mensajes publicados (número máximo accesible por la web). La intención es medir en un grupo de comunicaciones normales, que pretenden ser entendidas por un conjunto amplio de usuarios, como se comunican los emisores y si esa comunicación permite extraer alguna conclusión con utilidad esteganográfica. Se intenta evitar, por tanto, jerga propia de un conjunto reducido de usuarios y sí la expresión de mensajes que desean ser leídos por una mayoría pero que pueden tener las características o comodidades propias del canal donde se emiten. Se busca por tanto un modelo estadístico común de conducta con utilidad esteganográfica. Se implementa un crawler en lenguaje JAVA y se recopila esta información construyendo un corpus final de 319.381 frases y 4.029.257 palabras [16]. Para el etiquetado de este corpus se implementa un programa en lenguaje JAVA que utiliza el etiquetador TreeTagger [17] considerando un mensaje Twitter como una frase [16]. La primera característica interesante detectada es que sólo se consume en media 48,1214% de la capacidad total (67 caracteres por frase), con la presencia por tanto de frases cortas e información muy instantánea lo cual da muestra de la presencia mayoritaria de adverbios de lugar y tiempo. El 38,57% de las frases empiezan con @, un 6,32% con RT, acaban con emoticon un 6,21% y acaban con direcciones web un 23,23%. A la vista de este estudio, lingüísticamente más ampliado en [16], no parece factible en primera aproximación ocultación lingüística por procedimiento público, a excepción de las técnicas comentadas con anterioridad y ocultando menos información (al ser los mensajes twitter de pocas palabras).

Aunque al seleccionar un conjunto de usuarios públicos no se esperaba un número notorio de faltas de ortografías (que podría tener utilidad en esteganografía). En la siguiente tabla se recopilan algunas mediciones sobre faltas de ortografía donde se observa, al menos para éstas y en el corpus recopilado, que este tipo de técnicas basada en estos fallos no sería muy productiva desde el punto de vista de la esteganografía lingüística.

Estructura Punto+espacio+Término		Nº total 102.179
Mayus	91494	89,5428%
Minus	1654	1,6187‰
Número	847	0,8289%
Palabra+o+Palabra que empieza por 'o'	11 ocurrencias	
Palabra+y+Palabra que empieza por 'i'	42 ocurrencias	
Omisión de tildes en:	Ocurrencias palabras	Nº palabras 4.027.746
Palabra acabada en "cion"	1544	0,0383%
Palabra acabada en "sion"	674	0,0167%
Adverbio acabado en "mente" (que debería ir acentuado)	148	0,0036%

Fig. 3. Medición de algunas erratas y faltas de ortografías de fácil computación.

Podría ser interesante orientar la investigación en este tipo concreto de redes sociales (frases breves) al uso de gramáticas libres de contexto (CFGs) [18]. En el pasado no se consideró este tipo de estructuras desde un punto de vista práctico debido a la problemática de que para crear estegotextos con una cierta cohesión-coherencia se necesita la creación de gramáticas complejas (cuya automatización no es trivial) y diccionarios de gran volumen de palabras con categorizaciones semánticas. En el caso de twitter o similares este problema puede ser minimizado en gran medida. En twitter los mensajes son frases sueltas, sin necesidad alguna de que se mantenga una coherencia global con las frases publicadas anteriormente. De hecho en la práctica muchas de las frases no tienen nada que ver con sus antecesoras. Considerando esto sería factible profundizar en la investigación de reglas gramaticales

simples y la ocultación estar basada en la selección de palabras concretas (que se eligen de un conjunto W categorizado semánticamente y por PoS/probabilidad) para cada parte de la regla gramatical definida. En la práctica, es aunar los conocimientos en modelos N-GRAM, etiquetado lingüístico y sustitución de sinónimos según contexto para avanzar en un tipo evolucionado de CFG probabilística. Si esto fuera así, y a falta (en el momento de escribir este artículo) de una implementación real, la información a ocultar se centraría en las categorías lingüísticas más probables (nombres, verbos, adjetivos y adverbios en este orden). La capacidad de ocultación dependería del sistema de codificación elegido, si la codificación tiene el mismo peso para cada palabra a seleccionar hablaríamos de una capacidad de logW (base2) por término de la regla gramatical habilitada. Independientemente de otros criterios de aceptabilidad, un mensaje twitter podría ocultar como poco del orden de 3*logWi (base2) bits, siendo Wi el conjunto de palabras para cada elemento de la regla gramatical considerada, dado que una frase sencilla al menos podría tener un nombre, verbo y complemento.

Las gramáticas CFGs son muy efectivas esteganográficamente para este objetivo, una especie de adaptación de la herramienta NICETEXT [18] con las indicaciones resaltadas anteriormente. A falta de una implementación real, queda pendiente analizar si un detector sería capaz de inferir anomalías que detectaran la presencia de información oculta por estos procedimientos.

VI. CONCLUSIÓN

En el último año y medio se ha avanzado en el análisis de la seguridad de propuestas públicas de esteganografía lingüística en lengua española. Las cifras actuales indica que la capacidad de ocultación es baja y requiere del uso de al menos textos de centenas de palabras para ocultar una cantidad de unos 128 bits útiles para distribuir una clave criptográfica, una url, etc.

El uso masivo de las redes sociales hace interesante analizar la posibilidad de ocultación de información en texto digital en ellas. En la práctica, redes sociales como tuenti, facebook, twitter o similares por la inmediatez provoca que en media los mensajes intercambiados sean de decenas de palabras. Esta limitación hace que sin trocear o distribuir los estegotextos creados, en frases o otras unidades lingüísticas, los procedimientos esteganográficos analizados serían más apropiados en blogs y si acaso en comentarios algo extensos (centenas de palabras) en foros.

La posibilidad de utilizar esteganografía lingüítica en redes sociales con los procedimientos documentados, con una seguridad no basada en oscuridad, se complica bastante. En este orden se decide analizar cómo se expresa una comunidad de usuarios determinados en la red twitter y aunque aparecen elementos que permitirían crear algún modelo esteganográfico útil resulta difícil justificar su seguridad en una propuesta pública.

Se deja abierto al futuro nuevos avances para mejorar las técnicas de esteganografía lingüística y su posible adaptación a nuevos canales masivos de comunicación, como las redes sociales o cualquier medio futuro que dificultara la detección de comunicaciones subrepticias a un potencial estegoanalista.

REFERENCIAS

[1] J. Carracedo, Seguridad en Redes Telemáticas. Mc-Graw Hill. España. ISBN:84-481-4157-1 (2004), pág 123-131.

[2] F. L. Bauer. Decrypted Secrets. Methods and Maxims of Cryptology. s.l. : Springer; 1 edition (October 2, 1997), 1997. ISBN-13: 978-3540604181.

[3] D. Kahn, The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet. s.l. : Scribner; Rev Sub edition (December 5, 1996). ISBN-13: 978-0684831305.

[4] R. Bergmair, A comprehensive Bibliography of Linguistic Steganography. SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents, volume 6505, January 2007.

[5] A. Muñoz, J. Carracedo, S. Sánchez, Detection of distributed steganographic information in social networks. EATIS 2008. Euro American Conference on Telematics and Information Systems, September 10-12. Aracaju, Brazil. ACM-DL.

[6] A. Muñoz, J. Carracedo, StegSecret: Una herramienta de estegoanálisis pública. Mar del Plata : Congreso Iberoamericano de Seguridad de la Información. CIBSI 2007. http://stegsecret.sourceforge.net.

[7] A. Muñoz, J. Carracedo, Estegoanálisis aplicado a la generación automática de estegotextos en lengua española. CIBSI 2009. V Congreso Iberoamericano de Seguridad Informática. 16-18 Nov 2009. Montevideo-Uruguay.

[8] J. Greenberg, Some universals of grammar with particular reference to the order of meaningful elements. Cambridge, Mass: MIT Press. 1966

[9] A. Muñoz, I. Argüelles, J. Carracedo, Improving N-Gram linguistic steganography based on templates. International Conference on Security and Cryptography. Secrypt 2010. July 26-28 Athens, Greece.

[10] A. Muñoz, I. Argüelles, J. Carracedo, Hiding short secret messages based on linguistic steganography and manual annotation. Third IEEE International Symposium on Trust, Security and Privacy for Emerging Applications. TSP 2010. June 29-July 1, 2010 in Bradford, UK.

[11] A. Muñoz, I. Argüelles, J. Carracedo, Modificaciones sintácticas en lengua española con utilidad en esteganografía lingüística. RAEL-Revista Electrónica de Lingüística Aplicada. I.S.S.N.: 1885-9089 número 8. Fecha estimada publicación: abril 2010.

[12] A. Muñoz, I. Argüelles, J. Carracedo, Measuring the security of linguistic steganography in Spanish based on synonymous paraphrasing with WSD. Third IEEE International Symposium on Trust, Security and Privacy for Emerging Applications. TSP 2010. June 29-July 1, 2010 in Bradford, UK.

[13] M. Topkara, U. Topkara, M.J. Atallah, Information hiding through errors: a confusing approach, in: Proceedings of SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents, San Jose, CA, USA, January 29-February 1, 2007.

[14] J. Yago, Como sincronizar un comando terrorista con Twitter. 15/09/2008. http://www.securitybydefault.com/2008/09/como-sincronizar-un-comando-terrorista.html.

[15] R. Signel, Hackers Use Twitter to Control Botnet. August 13, 2009. http://www.wired.com/threatlevel/2009/08/botnet-tweets/

[16] A. Muñoz, I. Argüelles, Análisis de discurso en redes sociales. Twitter un caso bajo estudio. XXVIII Congreso Internacional de la Asociación Española de lingúística aplicada. AESLA 2010. Abril 2010. Vigo.

[17] H. Schmid, TreeTagger - a language independent part-of-speech tagger. Institute for Computational Linguistics of the University of Stuttgart. 2009. Available: http://www.ims.unistuttgart.de/projekte/corplex/TreeTagger/

[18] M. T. Chapman and G. I. Davida, Hiding the hidden: A software system for concealing ciphertext as innocuous text, in Information and Communications Security: First International Conference, Lecture Notes in Computer Science 1334, Springer, August 1997.

On the size of the colluder set in fingerprinting attacks

Maria Bras-Amorós and Albert Vico-Oton

***Abstract*—A classical attack to fingerprinting of digital contents is obtaining pirate copies by comparison of the licit copies of a set of colluders. A lot of research has been done for defining tracing algorithms for identifying at least one of the colluders that originated a given pirate copy. Our aim is to elaborate on the minimum number of colluders capable of generating a given pirate copy when the code used for fingerprinting is a Reed-Solomon code.**

Our main result is a lower bound on this minimum number. In the application side, having this lower bound means that once an illegal copy is caught, we can assert that at least a certain number of colluders, given by this bound, were involved in it. This result is illustrated with several examples showing that in many cases the bound is sharp.

The bound is extended to partially-erased fingerprints. The result on partially-erased fingerprints can be used in turn for bounding the number of colluders that were not caught once a subset of the colluders is caught.

***Index Terms*—Fingerprinting, polynomials over finite fields, Reed-Solomon codes**

I. Introduction

In the digital era one main concern is the illegal redistribution of digital contents. One way to fight it is by marking every single copy of the material that one does not want to have redistributed. This can be done by embedding a different imperceptible string of bits or symbols to each copy. Once an illegal copy is caught, if it was not modified, the illegal redistributor can be reidentified by the mark in his/her copy. This is called fingerprinting.

An attack to fingerprinting can be performed by a group of colluders. They can compare their copies and create a new pirate copy by erasing all the bits or symbols in which their copies differ or by using at each position where they differ, the bit or symbol that one of the users has there.

Formally, a subset of Σ^n for some alphabet Σ and positive integer n, called a code, is fixed. Then each depositary of a digital content is assigned a code word. A pirate copy is a vector $u = (u_0, \ldots, u_{n-1})$ in Σ^n, which is obtained from a set of colluders as follows. If the code words corresponding to the colluders are $c^{(1)}, \ldots, c^{(s)}$, then for all i in $\{0, \ldots, n-1\}$ one has $u_i = c_i^{(j)}$ for some j in $\{1, \ldots, s\}$, where $c_i^{(j)}$ is the ith coordinate of $c^{(j)}$. If erasures are also considered, then the pirate copy belongs to $(\Sigma \cup \{?\})^n$, and it must satisfy that for all i, either $u_i = c_i^{(j)}$ for some j or $u_i = ?$. If a pirate copy contains erasures we say that it is a shortened copy.

Reed-Solomon codes are a classical family of error control codes which have been extensively used also in the context of fingerprinting.

The identifiable parent property (IPP), for which all sets of colluders capable of generating a given pirate copy share at least one colluder, is defined in [1]. It is a desirable property when we are interested in the applications to fingerprinting. It is proved in the same reference that there exist Reed-Solomon codes with this property.

Another important property for fingerprinting codes is that given a pirate copy one of the colluders can always be identified by performing minimum distance error correction whenever the pirate copy has been created by at most a given number w of colluders. This property is denoted w-traceability or w-TA. Reed-Solomon codes are also used to find w-TA codes [2].

Further results on Reed-Solomon codes and the IPP and w-TA properties can be found in [3], [4]. Also generalized Reed-Solomon codes are used in [5] for dealing with shortened and corrupted fingerprints.

While the classical problem of fingerprinting is defining tracing algorithms for identifying at least one of the colluders that originated a given pirate copy, our aim is to elaborate on the minimum number of colluders capable of generating a given pirate copy when the code used for fingerprinting is a Reed-Solomon code.

Our main result (Theorem 2) is a lower bound on this minimum number. In the application side, having this lower bound means that once an illegal copy is caught, we can assert that at least a certain number of colluders, given by this bound, were involved in it. This result is illustrated with several examples showing that in many cases the bound is sharp.

The tools used for proving the main result are then used to prove an upper bound on the minimum number M of colluders that can obtain any given pirate copy. This same bound can be also derived from the fact that Reed-Solomon codes are MDS. Using the main result we then see that this upper bound is sharp which means that there are certain pirate copies which can not be obtained with fewer than M colluders.

The bound in Theorem 2 is extended to shortened fingerprints obtaining an analogous bound for this case. This result can be used in turn for bounding the number of colluders that were not caught once a subset of the colluders is caught.

Finally we point out the main drawback of this bound and sketch the way to overcoming it by a closer look to the interpolated polynomial. We finish with an open question whose solution would bring out a significant improvement of our bound.

II. Reed-Solomon codes and interpolating polynomials

Let $\mathbb{F}_q$ be the field with q elements (q a prime power) and let α be a primitive element of $\mathbb{F}_q$. Then $\mathbb{F}_q =$

$\{0, 1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\}$. The Reed-Solomon code of length $n = q-1$ and dimension k, denoted $RS_q(k)$, is the set

$$\{(f(1), f(\alpha), f(\alpha^2), \ldots, f(\alpha^{n-1})) : f \in \mathbb{F}_q[x], \deg(f) < k\}.$$

First we notice that for each vector $u = (u_0, \ldots, u_{n-1})$ in $\mathbb{F}_q^n$, there exists a unique polynomial f_u of degree at most $n-1$ such that $f_u(\alpha^i) = u_i$ for all i in $\{0, \ldots, n-1\}$. It can be computed, for instance, using the formula

$$f_u = \sum_{i=0}^{n-1} \left(u_i \prod_{\substack{j=0 \\ j \neq i}}^{n-1} \frac{x - \alpha^j}{\alpha^i - \alpha^j} \right).$$

The matrix of this system is a Vandermonde matrix which is known to be invertible. So, any u in $\mathbb{F}_q^n$ is of the form $(f(1), f(\alpha), f(\alpha^2), \ldots, f(\alpha^{n-1}))$ for some unique $f \in \mathbb{F}_q[x]$ of degree less than n.

Given a vector $u = (u_0, \ldots, u_{n-1})$ in $\mathbb{F}_q^n$, u is a code word if and only if $\deg(f_u) < k$. Now if $\deg(f_u) < k$ then u is a code word and so it can be obtained by just one single colluder. Our focus is on the case when $u \notin RS_q(k)$ and so when $\deg(f_u) \geq k$.

III. A LOWER BOUND

Lemma 1. *With the same notations as above, a vector $u = (u_0, \ldots, u_{n-1})$ in $\mathbb{F}_q^n$, $u \notin RS_q(k)$, agrees with any code word $c \in RS_q(k)$ in at most $\deg(f_u)$ positions.*

Proof: If the vector u agrees with a code word c in the position corresponding to the ith power of α this means that $f_u(\alpha^i) = f_c(\alpha^i)$ and so $(f_u - f_c)(\alpha^i) = 0$. Now since the number of roots of a polynomial over a finite field is upper bounded by its degree, the equality $(f_u - f_c)(\alpha^i) = 0$ will be satisfied by at most $\deg(f_u - f_c)$ different powers of α. But since $u \notin RS_q(k)$, $\deg(f_u) \geq k$ and since $c \in RS_q(k)$, $\deg(f_c) < k$. So, $\deg(f_u - f_c) = \deg(f_u)$ and $u = (u_0, \ldots, u_{n-1})$ agrees with c in at most $\deg(f_u)$ positions. ∎

Theorem 2. *With the same notations as above, the minimum number of colluders required to obtain a vector $u = (u_0, \ldots, u_{n-1})$ in $\mathbb{F}_q^n$, with $u \notin RS_q(k)$ is at least $\left\lceil \frac{n}{\deg(f_u)} \right\rceil$.*

Proof: It is a consequence of Lemma 1. ∎

Next we will illustrate this theorem with basic examples for $\deg(f_u)$ equal to $1, 2$ and $n-1$, and then with two further examples dealing with the norm and trace polynomials. See [6] for more details on these polynomials related to finite fields. We will see that for the examples with $\deg(f_u) = 1$ and most of the examples with $\deg(f_u) = 2$, and the examples with the norm and trace polynomials the lower bound is sharp. However, the example with $\deg(f_u) = n-1$ shows that the bound may be not sharp.

a) The case $\mathbf{deg(f_u) = 1}$*:* Suppose first that $\deg(f_u) = 1$. This case only makes sense when $k = 1$ and so $RS_q(k)$ is the repetition code. In this case $\lceil \frac{n}{\deg(f_u)} \rceil$ equals n. Since $\deg(f_u) = 1$ it holds that $f_u = a_0 + a_1 x$ for some $a_0, a_1 \in \mathbb{F}_q$, $a_1 \neq 0$ and so f_u represents a permutation of $\mathbb{F}_q$. The n different components of u can be covered by $n = q-1$ out of the q constant vectors of the repetition code.

b) The case $\mathbf{deg(f_u) = 2}$*:* The case $\deg(f_u) = 2$ makes sense when $k = 1$ or $k = 2$. In this case $\lceil \frac{n}{\deg(f_u)} \rceil$ equals $\frac{q-1}{2}$ if q is odd and $\frac{q}{2}$ if q is even. Since $\deg(f_u) = 2$ it holds that $f_u = a_0 + a_1 x + a_2 x^2$ for some $a_0, a_1, a_2 \in \mathbb{F}_q$, $a_2 \neq 0$. Consider the set $U = \{a_0 + a_1\beta + a_2\beta^2, \beta \in \mathbb{F}_q\}$. One can check that the equation on γ given by $a_0 + a_1\beta + a_2\beta^2 = a_0 + a_1\gamma + a_2\gamma^2$ has only two possible solutions, $\gamma = \beta$ and $\gamma = -\frac{a_1}{a_2} - \beta$. The two solutions are equal only for those β's such that $\beta = -\frac{a_1}{a_2} - \beta$. If q is odd this is only possible for $\beta = -\frac{a_1}{2a_2}$ so U has exactly $\frac{q-1}{2} + 1$ elements. Conversely, if q is even then $\beta = -\frac{a_1}{a_2} - \beta$ is true for all β if $a_1 = 0$ and it is false for all β if $a_1 \neq 0$. So, in the case q even the set U has either q elements if a_1 equals 0 and $\frac{q}{2}$ if $a_1 \neq 0$. Now, the set of components of u is exactly the set of elements in $U_0 = \{a_0 + a_1\beta + a_2\beta^2, \beta \in \mathbb{F}_q \setminus \{0\}\}$. Now, for q odd, $\#U_0 = \frac{q-1}{2}$ if $a_1 = 0$ and $\#U_0 = \frac{q-1}{2} + 1$ if $a_1 \neq 0$, while for q even, $\#U_0 = q-1$ if $a_1 = 0$ and $\#U_0 = \frac{q}{2}$ if $a_1 \neq 0$. So, if we take the repetition code, that is, $k = 1$ then the upper bound is tight only for q odd and $a_1 = 0$ or for q even and $a_1 \neq 0$. We will see in the next section that for $k = 2$ the bound is always tight.

c) The case $\mathbf{deg(f_u) = n-1}$*:* If $\deg(f_u) = n-1$ then the upper bound on the number of colluders is $\lceil \frac{n}{n-1} \rceil = 2$. But the only information given by this bound is that one single colluder could not obtain u. And this is already known from the fact that $u \notin RS_q(k)$.

d) Example with the trace polynomial: Suppose $q = \tilde{q}^m$ for some prime power $\tilde{q}$ and positive integer m. Then $\mathbb{F}_{\tilde{q}}$ is a subfield of $\mathbb{F}_q$. The trace polynomial of the field extension $\mathbb{F}_q/\mathbb{F}_{\tilde{q}}$, is defined as the polynomial

$$T(x) = x^{\tilde{q}^{m-1}} + x^{\tilde{q}^{m-2}} + \ldots + x^{\tilde{q}} + x.$$

The trace polynomial when evaluated at $\mathbb{F}_q$ gives elements of $\mathbb{F}_{\tilde{q}}$. The antiimage of each element in $\mathbb{F}_{\tilde{q}}$ consists of exactly $\tilde{q}^{m-1}$ elements in $\mathbb{F}_q$. All this means that if we take the pirate word $u = (u_0, \ldots, u_{n-1}) = (T(1), T(\alpha), T(\alpha^2), \ldots, T(\alpha^{n-1}))$ then u has for each β in $\mathbb{F}_{\tilde{q}} \setminus \{0\}$ exactly $\tilde{q}^{m-1}$ components equal to β plus $\tilde{q}^{m-1} - 1$ components equal to 0. In particular u can be obtained from the $\tilde{q}$ colluders consisting of the constant vectors $(\beta, \ldots, \beta)$ with $\beta \in \mathbb{F}_{\tilde{q}}$. These constant vectors are obtained by evaluating constant polynomials (with degree at most 0) in $1, \alpha, \ldots, \alpha^{n-1}$ and so they are code words of $RS_q(k)$ for any $k > 0$. So, u can be obtained by only $\tilde{q}$ colluders.

On the other hand, since $\deg(T) = \tilde{q}^{m-1} < q - 1 = n$, by the uniqueness of f_u we have that $f_u = T$ and so $\lceil \frac{n}{\deg(f_u)} \rceil = \lceil \frac{q-1}{\tilde{q}^{m-1}} \rceil = \lceil \frac{\tilde{q}^m - 1}{\tilde{q}^{m-1}} \rceil = \lceil \tilde{q} - \frac{1}{\tilde{q}^{m-1}} \rceil = \tilde{q}$. So, we can see that in this case the bound in Theorem 2 is sharp.

e) Example with the norm polynomial: We use now the same notations as before, just with the assumption that $\tilde{q} \neq 2$. The norm polynomial of the field extension $\mathbb{F}_q/\mathbb{F}_{\tilde{q}}$, is defined as the polynomial

$$N(x) = x^{\frac{\tilde{q}^m - 1}{\tilde{q} - 1}}.$$

The norm polynomial when evaluated at $\mathbb{F}_q$ gives also elements of $\mathbb{F}_{\tilde{q}}$. The antiimage of each element β in $\mathbb{F}_{\tilde{q}}$ consists of exactly $\frac{\tilde{q}^m-1}{\tilde{q}-1}$ elements in $\mathbb{F}_q$ if $\beta \neq 0$ and exactly one if $\beta = 0$. All this means that if we take the pirate word $u = (u_0, \ldots, u_{n-1}) = (N(1), N(\alpha), N(\alpha^2), \ldots, N(\alpha^{n-1}))$ then u has for each β in $\mathbb{F}_{\tilde{q}} \setminus \{0\}$ exactly $\frac{\tilde{q}^m-1}{\tilde{q}-1}$ components equal to β. In particular u can be obtained from the $\tilde{q}-1$ colluders consisting of the constant vectors $(\beta, \ldots, \beta)$ with $\beta \in \mathbb{F}_{\tilde{q}} \setminus \{0\}$ which, as explained before, are code words of $RS_q(k)$ for any $k > 0$. So, u can be obtained by only $\tilde{q}-1$ colluders.

On the other hand, since $\deg(N) = \frac{\tilde{q}^m-1}{\tilde{q}-1} < q-1 = n$, by the uniqueness of f_u we have that $f_u = N$ and so $\lceil \frac{n}{\deg(f_u)} \rceil = \lceil \frac{q-1}{\frac{\tilde{q}^m-1}{\tilde{q}-1}} \rceil = \lceil \tilde{q}-1 \rceil = \tilde{q}-1$. So, we can see that again in this case the bound in Theorem 2 is sharp.

IV. An upper bound, revisited

Theorem 3. *A number of $\lceil \frac{n}{k} \rceil$ colluders is enough for obtaining any pirate copy $u \in \mathbb{F}_q^n$.*

Proof: Consider a set of k positions $i_1, \ldots, i_k$ in u. The polynomial

$$f = \sum_{l=1}^{k} \left(u_{i_l} \prod_{\substack{j=1 \\ j \neq i_l}}^{k} \frac{x - \alpha^{i_j}}{\alpha^{i_l} - \alpha^{i_j}} \right)$$

has degree less than k and so $(f(1), f(\alpha), f(\alpha^2), \ldots, f(\alpha^{n-1}))$ is a code word c in $RS_q(k)$. Also $c_{i_l} = f(\alpha^{i_l}) = u_{i_l}$ for all l in $\{1, \ldots, k\}$. Then c and u agree in the positions $i_1, \ldots, i_k$. The same argument holds for any selection of less then k positions.

Divide u into $\lfloor \frac{n}{k} \rfloor$ disjoint sets of k positions plus the set of the $n - \lfloor \frac{n}{k} \rfloor$ remaining positions which are less than k and which may be empty. For each of these sets we can find a code word as before, which agrees with u in the selected positions. This gives a set of $\lceil \frac{n}{k} \rceil$ code words capable of generating the pirate copy u. ∎

We note that the proof of this theorem could have been much simplified by using the additional fact that Reed-Solomon codes are MDS. We chose this proof because it only uses the same tools used for the previous theorem.

From Theorem 2 we deduce that the bound on the number of colluders in Theorem 3 is attained when f_u has degree exactly equal to k. So, there are words in $\mathbb{F}_q^n$ which can not be obtained by fewer than $\lceil \frac{n}{k} \rceil$ colluders.

V. On shortened fingerprints

In this section we make some remarks on the case when part of the fingerprint has been simply erased. Tracing of traitors based on shortened fingerprints is treated in [5]. We show that the results in the previous sections can be naturally extended to this case.

Formally, instead of considering pirate copies in $\mathbb{F}_q^n$ we consider pirate copies $u = (u_0, \ldots, u_{n-1})$ in $(\mathbb{F}_q \cup \{?\})^n$ which are obtained from a set of colluders as follows. If the code words corresponding to the colluders are $c^{(1)}, \ldots, c^{(s)}$, then for all i in $\{0, \ldots, n-1\}$ one has $u_i = c_i^{(j)}$ for some j in $\{1, \ldots, s\}$, or $u_i = ?$. We call n^* the number of erased positions in u, that is, the number of components u_i which are equal to ?. Now $n - n^*$ will play the role played by n in the previous sections.

The polynomial f_u can be redefined as follows.

$$f_u = \sum_{\substack{i=0 \\ u_i \neq ?}}^{n-1} \left(u_i \prod_{\substack{j=0 \\ u_j \neq ? \\ j \neq i}}^{n-1} \frac{x - \alpha^j}{\alpha^i - \alpha^j} \right).$$

The vector $(f_u(1), f_u(\alpha), \ldots, f_u(\alpha^{n-1}))$ has the particularity that it agrees with u in all the non-erased positions and that it is the one with smallest degree with this property. Notice that now the degree of f_u is at most $n - n^* - 1$ and that f_u is, as before, the unique polynomial agreeing with u in all the non-erased positions and with degree at most $n - n^* - 1$. Uniqueness follows as before using a Vandermonde matrix. If $n^* = 0$ then the polynomial f_u is the same polynomial that we already had.

If $\deg(f_u) < k$ and in particular, if $n - n^* - 1 < k$ then $(f_u(1), f_u(\alpha), \ldots, f_u(\alpha^{n-1}))$ is a code word and so u agrees with a code word in all its non-erased positions. So, it can be obtained with just one colluder. Next we consider the case $\deg(f_u) \geq k$.

Lemma 1 and Theorem 2 can be now reformulated. The proof of the new lemma is parallel to that of Lemma 1 and hence it has been omitted. Also, the proof of the new theorem follows from the lemma.

Lemma 4. *A vector u in $(\mathbb{F}_q \cup \{?\})^n$, $u \notin RS_q(k)$, agrees with any code word $c \in RS_q(k)$ in at most $\deg(f_u)$ positions.*

Theorem 5. *Suppose that a vector u is in $(\mathbb{F}_q \cup \{?\})^n$, and $u \notin RS_q(k)$. Then the minimum number of colluders required to obtain u is at least $\left\lceil \frac{n-n^*}{\deg(f_u)} \right\rceil$.*

We would like to end with the remark that this result can be used in turn for bounding the number of colluders that were not caught once a subset of the colluders is caught. Indeed, suppose that a set of colluders is caught that collaborated in the pirate copy u. Then erase all the positions of the pirate copy which agree with the copy of at least one of the caught colluders and obtain a new pirate copy u^*. Let n^{**} be the total number of erased positions in u^*. Then, Theorem 5 applied to u^* tells us that at least $\left\lceil \frac{n-n^{**}}{\deg(f_{u^*})} \right\rceil$ colluders are still not caught.

VI. Drawback and overcoming it by a closer look to polynomial factorization

In Table I there is an analysis of the performance of the bound in Theorem 2. It turns out that $\lceil \frac{n}{\deg(f_u)} \rceil$ is most of the times 2 and this does not introduce any information. Indeed, having at least two colluders is equivalent to having u not a code word, which is something that is very easy to check without any need of interpolating polynomials. The

$k=3$	coalition size	trials	mean value m of $\deg(f_u)$	$\lceil \frac{n}{m} \rceil$
	2	100	24.95	2
	3	100	24.99	2
	4	100	24.95	2
	5	100	24.95	2
	6	100	24.95	2
	7	100	25.00	2
	8	100	24.98	2
	9	100	24.94	2
$k=4$	coalition size	trials	mean value m of $\deg(f_u)$	$\lceil \frac{n}{m} \rceil$
	2	100	24.96	2
	3	100	24.96	2
	4	100	24.98	2
	5	100	24.97	2
	6	100	24.96	2
	7	100	24.96	2
$k=5$	coalition size	trials	mean value m of $\deg(f_u)$	$\lceil \frac{n}{m} \rceil$
	2	100	24.97	2
	3	100	24.98	2
	4	100	24.97	2
	5	100	24.97	2
	6	100	24.96	2

TABLE I
WE CONSIDER $RS_{27}(3)$, $RS_{27}(4)$, AND $RS_{27}(5)$. FOR EACH OF THESE CODES AND FOR EACH COALITION SIZE s FROM $s=2$ TO $s=\lceil \frac{n}{k} \rceil$, WE GENERATED 100 PIRATE COPIES u, EACH ONE RANDOMLY OBTAINED FROM s RANDOM COLLUDERS. WE COMPUTED THE MEAN m OF $\deg(f_u)$ AMONG THESE 100 PIRATE COPIES. THEN WE COMPUTED $\lceil \frac{n}{m} \rceil$, WHICH SHOULD GIVE AN IDEA OF WHAT WE CAN EXPECT FROM THE BOUND IN THEOREM 2.

reason for having $\lceil \frac{n}{\deg(f_u)} \rceil = 2$ most of the times is that having $\lceil \frac{n}{\deg(f_u)} \rceil > 2$ would mean $\deg(f_u) < \frac{n}{2}$ and, if $f_u = a_0 + a_1 x + \ldots + a_{n-1} x^{n-1}$, this would mean having $a_{\lceil \frac{n}{2} \rceil} = \ldots = a_{n-1} = 0$, which happens only with a probability $\frac{q^{\lceil \frac{n}{2} \rceil - 1}}{q^n} = q^{-\lceil \frac{n}{2} \rceil - 1}$.

The limitation of this bound already comes from Lemma 1 and the fact that, in its proof, we only bound the number of roots of the polynomial $f_u - f_c$ (and so the number of agreements between the caught word u and a general code word c) by the degree of $f_u - f_c$ which is exactly the degree of f_u. But conversely, it is well known that a random polynomial from $\mathbb{F}_q[x]$ has "on the average", as q increases, exactly one root in the field $\mathbb{F}_q$ [7], [8], [9]. So, on average, our bound in Lemma 1 is very far from the real number of agreements between the caught word and a general code word.

But we have more information on the polynomial $f_u - f_c$ rather than its degree. Indeed we know all its terms of degree at least k. In some cases this knowledge may slightly modify the result in Lemma 1 and this may improve drastically the bound in Theorem 2.

In the next lemma we illustrate it with a particular example.

Lemma 6. *Suppose that $q = \tilde{q}^m$ for some prime power $\tilde{q}$ and positive integer m. Suppose that $u \in \mathbb{F}_q^n$, $u \notin RS_q(k)$ with $1 \leq k \leq \tilde{q}^{m-1}$ is such that $f_u = x^{\tilde{q}^{m-1}} + x^{\tilde{q}^{m-2}} + \ldots + x^{\tilde{q}^i} + g_k(x)$, with $\tilde{q}^{i-1} < k < \tilde{q}^i$ and $g_k(x)$ a polynomial of degree less than k. Then the word u agrees with any code word $c \in RS_q(k)$ in at most $(k-1)\tilde{q}$ positions.*

Proof: Suppose that u and c agree in the position corresponding to α^i and suppose that $T(\alpha^i) = \beta \in \mathbb{F}_{\tilde{q}}$. Then α^i is a root of the polynomial

$$h_\beta(x) = f_u(x) - f_c(x) - T(x) + \beta.$$

In general, all the powers α^i corresponding to positions where u and c agree are roots of $h_\beta(x)$ for some $\beta \in \mathbb{F}_{\tilde{q}}$, and so they are roots of

$$H(x) = \prod_{\beta \in \mathbb{F}_{\tilde{q}}} h_\beta(x).$$

By the hypothesis on f_u, $\deg(f_u - T(x)) < k$ and also $\deg(f_c) < k$ and $\deg(\beta) \leq 0 < k$. So, $\deg(h_\beta) < k$ for all $\beta \in \mathbb{F}_{\tilde{q}}$ and $\deg(H) \leq (k-1)\tilde{q}$. This proves the Lemma. ∎

Lemma 6 leads to a refinement of the bound in Theorem 2 for this particular case, whenever $(k-1)\tilde{q} < \deg(f_u)$. Indeed the new bound is $\lceil \frac{n}{(k-1)\tilde{q}} \rceil$.

An analogous lemma associated to the norm polynomial is presented next. The proof is left to the reader since it is very similar to the previous one. The only difference is on the fact that the norm polynomial evaluated at powers of α runs $\mathbb{F}_{\tilde{q}} \setminus \{0\}$ while the trace polynomial runs all $\mathbb{F}_{\tilde{q}}$.

Lemma 7. *Suppose that $q = \tilde{q}^m$ for some prime power $\tilde{q}$ and positive integer m. Suppose that $u \in \mathbb{F}_q^n$, $u \notin RS_q(k)$ with $1 \leq k \leq \frac{\tilde{q}^m - 1}{\tilde{q} - 1}$ is such that $f_u = x^{\frac{\tilde{q}^m - 1}{\tilde{q} - 1}} + g_k(x)$, with $g_k(x)$ a polynomial of degree less than k. Then the word u agrees with any code word $c \in RS_q(k)$ in at most $(k-1)(\tilde{q}-1)$ positions.*

In this particular case the bound in Theorem 2 would be improved to $\lceil \frac{n}{(k-1)(\tilde{q}-1)} \rceil$.

We leave it as an open question to have an equivalent of these lemmas for general polynomials. That is, given a polynomial $f(x)$ we wish to have an upper bound on the number of roots of the polynomial $x^k f(x) + g_k(x)$, with $\deg(g_k(x)) < k$, depending only on k and $f(x)$. This is obviously solved for the case of a constant polynomial $f(x)$ and in this case the bound is exactly k. For the general case we would like the bound to be smaller than $k + \deg(f)$.

VII. CONCLUSION

We used some of the very special properties of Reed-Solomon codes to bound the minimum size of a collusion party, once a pirate copy built from a set of colluders is caught. Some experimental research brought to light a drawback of our bound. We elaborated on this drawback and gave some solutions for particular cases. This lead to an open question about polynomial factorization over finite fields for polynomials whose larger order coefficients are known.

ACKNOWLEDGMENT

The authors are grateful to Marcel Fernández and Josep Domingo-Ferrer for many helpful comments. This work was partly supported by the Spanish Government through projects TIN2009-11689 "RIPUP" and CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", and by the Government of Catalonia under grant 2009 SGR 1135.

REFERENCES

[1] H. D. L. Hollmann, J. H. van Lint, J.-P. M. G. Linnartz, and L. M. G. M. Tolhuizen, "On codes with the identifiable parent property," *J. Comb. Theory, Ser. A*, vol. 82, no. 2, pp. 121–133, 1998.

[2] J. Staddon, D. R. Stinson, and R. Wei, "Combinatorial properties of frameproof and traceability codes," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 1042–1049, 2001.

[3] A. Silverberg, J. Staddon, and J. L. Walker, "Applications of list decoding to tracing traitors," *IEEE Trans. Inform. Theory*, vol. 49, no. 5, pp. 1312–1318, 2003.

[4] M. Fernandez, J. Cotrina, M. Soriano, and N. Domingo, "A note about the identifier parent property in Reed-Solomon codes," *Computers & Security*, vol. In Press, Corrected Proof, pp. –, 2010.

[5] R. Safavi-Naini and Y. Wang, "Traitor tracing for shortened and corrupted fingerprints," in *Digital Rights Management Workshop*, ser. Lecture Notes in Computer Science, Springer, Ed., vol. 2696, 2003, pp. 81–100.

[6] R. Lidl and H. Niederreiter, *Finite fields*, 2nd ed., ser. Encyclopedia of Mathematics and its Applications. Cambridge: Cambridge University Press, 1997, vol. 20, with a foreword by P. M. Cohn.

[7] P. Flajolet, X. Gourdon, and D. Panario, "The complete analysis of a polynomial factorization algorithm over finite fields," *J. Algorithms*, vol. 40, no. 1, pp. 37–81, 2001. [Online]. Available: http://dx.doi.org/10.1006/jagm.2001.1158

[8] D. Panario, "What do random polynomials over finite fields look like?" in *Finite fields and applications*, ser. Lecture Notes in Comput. Sci. Berlin: Springer, 2004, vol. 2948, pp. 89–108.

[9] V. K. Leont'ev, "On the roots of random polynomials over a finite field," *Mat. Zametki*, vol. 80, no. 2, pp. 313–316, 2006. [Online]. Available: http://dx.doi.org/10.1007/s11006-006-0139-y

Propiedades de trazabilidad de los códigos de Reed-Solomon para ciertos tamaños de coalición

José Moreira
Departamento de Ingeniería Telemática
Universitat Politècnica de Catalunya
Email: jose.moreira@entel.upc.edu

Marcel Fernández
Departamento de Ingeniería Telemática
Universitat Politècnica de Catalunya
Email: marcel@entel.upc.edu

Miguel Soriano
Departamento de Ingeniería Telemática
Universitat Politècnica de Catalunya
Email: soriano@entel.upc.edu

***Abstract*—Los códigos de fingerprinting se utilizan para disuadir la redistribución de contenidos digitales por usuarios legales pero deshonestos (traidores). En este contexto, los códigos con la propiedad de trazabilidad (TA) tienen una importancia destacada, dado que permiten identificar eficientemente, por lo menos, a uno de los traidores que ha participado en la redistribución. Los códigos con la propiedad identificadora de padres (IPP) también tienen esta capacidad de identificación, pese a que no poseen, en general, un algoritmo eficiente de identificación. Otros códigos que tienen una capacidad de identificación más *débil* son los códigos seguros contra incriminaciones (SFP). Es un resultado conocido que un código TA es un código IPP y que un código IPP es un código SFP. Sin embargo, las implicaciones en sentido contrario no son ciertas en general. Pese a eso, se sospecha que en el caso de los códigos de Reed-Solomon estas tres propiedades son equivalentes. En este artículo se investiga esta equivalencia y se proporciona una respuesta afirmativa para familias de códigos de Reed-Solomon en los que el número máximo de traidores divide al tamaño del cuerpo del código.**

I. Introducción

Dentro del campo de la distribución de contenidos digitales, la técnica del *fingerprinting* aparece como una posible solución para disuadir a usuarios deshonestos (traidores) de redistribuir copias de un contenido que han obtenido de forma legal. Con este fin, el distribuidor del contenido incrusta una marca, llamada huella digital o *fingerprint*, en cada una de las copias que va a distribuir. Este proceso debe aunar dos propiedades: primero, la inserción de la marca en el contenido debe realizarse de forma robusta e imperceptible para los usuarios; y segundo, debe insertarse una marca única para cada uno de ellos. Esto hace que cada usuario posea una copia del contenido personalizada, única, hecho que le disuade de distribuir su propia copia, puesto que de hacerlo, será fácilmente identificado.

No obstante, existe un punto débil en este esquema. Un grupo de traidores puede formar una coalición, comparar cada una de sus copias y determinar en qué partes son diferentes. Obviamente, las posiciones del contenido en las que difieren sus copias son posiciones en las que sus respectivas huellas digitales también difieren. De esta forma, los traidores pueden crear una nueva copia del contenido (copia pirata) alterando estas posiciones detectadas con el fin de que la copia pirata contenga una marca híbrida que no permita identificar a ninguno de los participantes de la coalición. Es más, podrían llegar a crear una copia pirata que contuviese la huella digital de algún usuario inocente, o fuese muy similar a ésta.

Llamamos al conjunto de marcas de usuario código de *fingerprinting*. Este código debe construirse de forma que, por lo menos, se garantice la protección de usuarios inocentes frente a una falsa incriminación. Un esquema más robusto debe permitir, además, identificar a usuarios traidores. Los códigos empleados en *fingerprinting* se clasifican en función de estas capacidades para proteger a usuarios inocentes o identificar a traidores. Estas propiedades no son equivalentes en general. Habitualmente se requieren condiciones más restrictivas para identificar a traidores que para proteger a usuarios inocentes. Sin embargo, existe la conjetura de que para el caso de los códigos de Reed-Solomomon estas propiedades (llamadas en conjunto propiedades de trazabilidad) se cumplen simultáneamente y no se pueden disociar. En este artículo se investiga esta conjetura, planteada en [1], [2].

El artículo se organiza de la siguiente manera. En la siguiente sección, introducimos el tema y presentamos la notación que se utilizará. En la sección III presentamos el resultado principal del artículo, demostrando que las propiedades de trazabilidad son equivalentes para determinadas familias de códigos de Reed-Solomon. También presentamos un algoritmo para encontrar conjuntos de marcas en el código que demuestran esta equivalencia. En la sección IV ilustramos el funcionamiento del algoritmo mediante un ejemplo. En la sección V comentamos la equivalencia de las propiedades para otros tamaños de coalición y finalmente presentamos las conclusiones del artículo.

II. Preliminares

Dado el cuerpo finito de q elementos, $\mathbb{F}_q$, denotamos los vectores de n coordenadas sobre $\mathbb{F}_q$ en negrilla, por ejemplo, $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}_q^n$. En particular, $\mathbf{1} = (1, \ldots, 1)$. Denotamos la distancia (de Hamming) entre dos vectores $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^n$ por $\mathrm{d}(\mathbf{a}, \mathbf{b})$.

Un (n, M, d)-código $\mathcal{C}$ sobre $\mathbb{F}_q$ es un subconjunto de $\mathbb{F}_q^n$ de tamaño M tal que la distancia mínima entre sus elementos es d. Habitualmente nos referiremos a los elementos de $\mathcal{C}$ como palabras código. Si $\mathcal{C}$ es un $\mathbb{F}_q$-espacio vectorial de dimensión k, diremos que $\mathcal{C}$ es un $[n, k, d]$-código.

Consideraremos que el conjunto de marcas que el distribuidor inserta en las copias antes de distribuirlas es un

código $\mathcal{C} \subseteq \mathbb{F}_q^n$. Cada $\mathbf{t} \in \mathcal{C}$ identifica unívocamente a un usuario. En adelante, designaremos por $\mathbf{t}$ indistintamente tanto la marca asociada a un usuario como al usuario en sí.

Como se ha comentado antes, si una coalición de c usuarios deshonestos compara sus copias podrán detectar un cierto número de posiciones donde sus copias, y por lo tanto sus marcas, difieren. Asumiremos que una coalición sólo puede crear copias pirata en las que solo estén alteradas las posiciones detectadas. Esto se conoce habitualmente en la literatura como *marking assumption* [3].

Definición 1. *Sea $\mathcal{C}$ un (n, M, d)-código sobre $\mathbb{F}_q$ y T un subconjunto de $\mathcal{C}$ de tamaño c, $T = \{\mathbf{t}^1, \dots, \mathbf{t}^c\}$. Diremos que $\mathbf{x} \in \mathbb{F}_q^n$ es un descendiente de T si para cada coordenada $1 \leq i \leq n$ existe un j, $1 \leq j \leq c$, tal que $x_i = t_i^j$. Diremos también que $\mathbf{t}^j$ es un padre de $\mathbf{x}$. El conjunto de todos los posibles descendientes de T se denota por* $\mathrm{desc}(T)$.

También asumiremos, como en [1], [2], [4]–[6], que el conjunto de copias pirata que una coalición T puede generar es $\mathrm{desc}(T)$. Denotaremos por $\mathrm{desc}_c(\mathcal{C})$ el conjunto de todos los descendientes que pueden ser generados por cualquier subconjunto $T \subseteq \mathcal{C}$ de tamaño máximo c. Si suponemos que en nuestro esquema de distribución de contenidos no se producirán coaliciones de tamaño mayor que c entonces, el conjunto de marcas pirata que existirá será precisamente $\mathrm{desc}_c(\mathcal{C})$.

Definición 2. *Un (n, M, d)-código $\mathcal{C}$ es (c_1, c_2)-seguro contra incriminaciones (SFP, de la denominación inglesa* secure frameproof*) si cualquier par de subconjuntos disjuntos $T_1, T_2 \subseteq \mathcal{C}$ de tamaño máximo c_1 y c_2 respectivamente satisfacen que sus conjuntos de descendientes son también disjuntos.*

Es decir, utilizando un código (c_1, c_2)-SFP (véase [2], [7]), ninguna coalición de tamaño c_1 podrá generar en el contenido pirata la marca que cualquier otro conjunto disjunto de hasta c_2 usuarios hubiese podido generar, y viceversa.

Definición 3. *Un (n, M, d)-código $\mathcal{C}$ tiene la propiedad c-identificadora de padres (IPP, de la denominación inglesa* identifiable parent property*) si para cualquier $\mathbf{x} \in \mathbb{F}_q^n$ o se cumple que $\mathbf{x} \notin \mathrm{desc}_c(C)$, o*

$$\bigcap_{\substack{T:\mathbf{x}\in\mathrm{desc}(T)\\ |T|\leq c}} T \neq \emptyset.$$

En otras palabras, utilizando un código c-IPP, la intersección de todas las coaliciones de tamaño máximo c capaces de generar $\mathbf{x}$ no es nula. En particular, las palabras código que pertenecen a la intersección corresponden a usuarios que han participado en la construcción del contenido pirata y por lo tanto, pueden ser identificados.

Definición 4. *Un (n, M, d)-código $\mathcal{C}$ tiene la propiedad de c-trazabilidad (TA) si para cualquier subconjunto $T \subseteq \mathcal{C}$ de tamaño máximo c satisface que $\mathrm{d}(\mathbf{x}, \mathbf{t}) < \mathrm{d}(\mathbf{x}, \mathbf{y})$ para algún $\mathbf{t} \in T$ y cualquier $\mathbf{y} \in \mathcal{C} \setminus T$.*

Dada una palabra pirata, ambas familias de códigos IPP y TA permiten identificar, por lo menos, a uno de los traidores. La principal diferencia es que el proceso de identificación se ejecuta en tiempo $O(\binom{M}{c})$ para códigos IPP mientras que para códigos TA se reduce a $O(M)$. Es fácil ver que la propiedad TA es más fuerte que la propiedad IPP [5].

Definición 5. *Dado un (n, M, d)-código $\mathcal{C}$ sobre $\mathbb{F}_q$ y dos subconjuntos de éste, $V, W \subseteq \mathcal{C}$, $V = \{\mathbf{v}^1, \mathbf{v}^2, \dots\}$, $W = \{\mathbf{w}^1, \mathbf{w}^2, \dots\}$, definimos la separación de grupo entre V y W, $\mathrm{D}(V, W)$, como el número de coordenadas donde los elementos de V y W contienen elementos disjuntos, es decir*

$$\mathrm{D}(V, W) = |\{i : \{v_i^1, v_i^2, \dots\} \cap \{w_i^1, w_i^2, \dots\} = \emptyset\}|.$$

Sea $\mathcal{C}$ un (n, M, d)-código y dados dos enteros c_1, c_2 denotamos por D_{c_1,c_2} el mínimo valor de $\mathrm{D}(T_1, T_2)$ entre cualquier par de subconjuntos disjuntos $T_1, T_2 \subseteq \mathcal{C}$ de tamaño máximo c_1 y c_2 respectivamente. Ciertamente, $D_{1,1} = d$. Además, diremos que T_1 y T_2 forman una (c_1, c_2)-configuración no separada si se cumple que $\mathrm{D}(T_1, T_2) = 0$.

Lema 6. *Para cualquier $[n, k, d]$-código $\mathcal{C}$ y cualquier par de valores enteros c_1, c_2 se satisface que*

$$\max\{0, d - (c_1c_2 - 1)(n - d)\} \leq D_{c_1,c_2} \leq \max\{0, d - (c_1 + c_2 - 2)(k - 1)\}.$$

Demostración: Es inmediata considerando que el lema es una generalización de [6, Lema 2.3]. ■

Obviamente, para un $[n, k, d]$-código si $d \leq (c_1+c_2-2)(k-1)$ el código no es (c_1, c_2)-SFP y si $d > (c_1c_2 - 1)(n - d)$ el código es (c_1, c_2)-SFP. En realidad, si $c_1 = c_2 = c$ se obtiene que esta última condición puede traducirse en $d > (1-1/c^2)n$, y en este caso, es un resultado conocido, que el código no es sólo (c, c)-SFP sino que también es c-IPP y c-TA. En general, para cualquier (n, M, d)-código tenemos que [5]

$$d > (1 - 1/c^2)n \Rightarrow c\text{-TA} \Rightarrow c\text{-IPP} \Rightarrow (c, c)\text{-SFP}. \quad (1)$$

Un (n, M, d)-código es MDS si cumple con igualdad la cota de Singleton, $M \leq q^{n-d+1}$. En particular, los códigos MDS lineales satisfacen que $n-d = k-1$. Para este tipo de códigos se puede ver [6] que la primera de las implicaciones en (1) se cumple en sentido inverso. Es decir, todo código MDS lineal es c-TA si y solo si $d > (1 - 1/c^2)n$.

Los códigos de Reed-Solomon son un tipo de códigos MDS lineales que, además, son cíclicos. Denotaremos por $\mathbf{t}^{(i)}$ la rotación cíclica de $\mathbf{t} \in \mathbb{F}_q^n$ en i coordenadas hacia la derecha.

Definición 7. *Sea α un elemento primitivo de $\mathbb{F}_q$. El código de Reed-Solomon de longitud $n = q-1$ y dimensión k,* $\mathrm{RS}(n, k)$*, se define como el conjunto*

$$\mathrm{RS}(n, k) = \{(f(1), f(\alpha), f(\alpha^2), \dots, f(\alpha^{q-2})) : f(x) \in \mathbb{F}_q[x]_{k-1}\},$$

donde $\mathbb{F}_q[x]_{k-1}$ denota el anillo de polinomios sobre $\mathbb{F}_q$ de grado máximo $k - 1$.

Si $f(x) \in \mathbb{F}_q[x]_{k-1}$, diremos que $f(x)$ genera la palabra código $\mathbf{t} = (f(1), \ldots, f(\alpha^{q-2})) \subseteq \mathrm{RS}(n,k)$.

En [1], [2] se planteó la cuestión de si es cierto que todos los códigos Reed-Solomon que cumplen la propiedad c-IPP también cumplen la propiedad c-TA. Sin embargo, para un gran número de códigos de Reed-Solomon resulta que no sólo c-IPP implica c-TA, sino que se satisface

$$d > (1 - 1/c^2)n \Leftrightarrow c\text{-TA} \Leftrightarrow c\text{-IPP} \Leftrightarrow (c,c)\text{-SFP}. \quad (2)$$

En otras palabras, ya sabemos que si $d > (1 - 1/c^2)n$ no existen (c,c)-configuraciones no separadas. La cuestión planteada consiste en encontrar por lo menos una (c,c)-configuración no separada cuando $d \leq (1 - 1/c^2)n$.

Por último, describimos aquí una última convención que tomaremos. Dado un polinomio $f(x) \in \mathbb{F}_q[x]$ podemos asociarle la aplicación $f : \mathbb{F}_q \to \mathbb{F}_q$ definida como $x \mapsto f(x)$. Por abuso del lenguaje, nos referiremos a esta aplicación simplemente como f.

III. Equivalencia de las propiedades de trazabilidad de los códigos de Reed-Solomon

En este apartado presentamos el resultado principal de este artículo, que se resume en el siguiente teorema.

Teorema 8. *Sea* $\mathrm{RS}(n,k)$ *un código de Reed-Solomon sobre* $\mathbb{F}_q$ *y* c *un divisor de* q*. Entonces, si la distancia mínima de* $\mathrm{RS}(n,k)$ *satisface* $d \leq n - n/c^2$ *el código no es* (c,c)*-SFP.*

Antes de presentar la demostración del teorema, introducimos un algoritmo para encontrar (c,c)-configuraciones no separadas en códigos de Reed-Solomon sobre $\mathbb{F}_q$ cuando se satisfacen las condiciones del Teorema 8. La demostración del teorema anterior seguirá el esquema del algoritmo.

A. Algoritmo

- Entrada: Un código de Reed-Solomon sobre $\mathbb{F}_q$, $\mathrm{RS}(n,k)$, con distancia mínima $d \leq (1 - 1/c^2)n$ y un entero c divisor de q.
- Salida: Un par de subconjuntos $T_1, T_2 \subseteq \mathrm{RS}(n,k)$ tales que forman una (c,c)-configuración no separada.

1) Si $c^2 > q$ entonces:
   a) Tomar $c' = \min\{c, n\}$.
   b) Tomar un polinomio arbitrario de primer grado $f(x) \in \mathbb{F}_q[x]_{k-1}$ y la palabra código que genera, $\mathbf{t} = (t_1, \ldots, t_n)$.
   c) Retornar
   $$T_1 = \{t_i \mathbf{1} : 1 \leq i \leq c'\} \text{ y}$$
   $$T_2 = \{\mathbf{t}^{(jc')} : 0 \leq j \leq \lceil n/c' \rceil - 1\}.$$
2) Si $c^2 \leq q$ entonces:
   a) Tomar un subgrupo aditivo $G \leq \mathbb{F}_q$ de q/c^2 elementos.
   b) Construir un polinomio no trivial de grado mínimo $f(x) \in \mathbb{F}_q[x]_{k-1}$ que tenga como raíces de multiplicidad 1 los elementos de G (la aplicación f asociada a $f(x)$ será un homomorfismo con imagen de tamaño c^2).
   c) Tomar un subgrupo $S \leq \operatorname{im} f$ de c elementos y sus c clases laterales, $\beta_1 + S, \ldots, \beta_c + S$.
   d) Tomar un valor aleatorio r, $1 \leq r \leq c$.
   e) Retornar
   $$T_1 = \{\beta_i \mathbf{1} : \beta_i \in \beta_r + S\} \text{ y}$$
   $$T_2 = \{\mathbf{t}^j : 1 \leq j \leq c\},$$
   donde $\mathbf{t}^j$ es la palabra código generada por el polinomio $f_j(x) = f(x) - \beta_j$, $1 \leq j \leq c$.

B. Correctitud del algoritmo

El algoritmo considera dos supuestos en función del valor de c. El primer supuesto, $c^2 > q$, queda demostrado en la siguiente proposición.

Proposición 9. *Sea* $\mathrm{RS}(n,k)$ *un código de Reed-Solomon sobre* $\mathbb{F}_q$ *y* c *un entero tal que* $c^2 \geq q - 1$*. Entonces, si la distancia mínima de* $\mathrm{RS}(n,k)$ *satisface que* $d \leq n - n/c^2$ *el código no es* (c,c)*-SFP.*

Demostración: Dado que se se cumple que el código es MDS, $n = q - 1$ y considerando las restricciones del enunciado, es inmediato comprobar que $k \geq 2$. Es decir, el código contiene, por lo menos, palabras constantes y palabras generadas a partir de polinomios de grado 1. Tomemos un polinomio arbitrario de grado 1, $f(x)$. La palabra que genera, $\mathbf{t}$, será tal que sus n coordenadas serán todas diferentes. Consideremos el conjunto de sus primeras $c' = \min\{c, n\}$ coordenadas, $C' = \{t_1, \ldots, t_{c'}\}$. Tomemos ahora rotaciones de la palabra $\mathbf{t}$ hacia la derecha, en múltiplos de c' coordenadas: $\mathbf{t}, \mathbf{t}^{(c')}, \mathbf{t}^{(2c')}, \mathbf{t}^{(3c')}, \ldots, \mathbf{t}^{(mc')}$. Es inmediato ver que si se toma el conjunto de palabras T_2 definido en el algoritmo, para cada coordenada, por lo menos una de las palabras de T_2 toma un valor en C'. Además $|T_2| = \lceil n/c' \rceil \leq c$.

Dado que el código contiene todas las palabras constantes, construimos T_1, de tamaño $c' \leq c$, como el conjunto de palabras constantes para cada uno de los elementos de C'. Así, para cada coordenada existe por lo menos una palabra en T_1 y una palabra en T_2 que comparten el mismo valor. Entonces T_1, T_2 es una (c,c)-configuración no separada. ■

Para demostrar el segundo supuesto, $c^2 \leq q$, necesitamos demostrar antes los siguientes lemas auxiliares.

Lema 10. *Sea* R *un subgrupo aditivo de tamaño* r *del cuerpo finito* $\mathbb{F}_q$*,* $R \leq \mathbb{F}_q$*. Entonces, si* m *divide a* r *existe un subgrupo* $S \leq R$ *con* $|S| = m$ *elementos.*

Demostración: El subgrupo S existe como consecuencia de los teoremas de Sylow [8]. ■

Lema 11. *Dado el cuerpo finito* $\mathbb{F}_q$ *y un entero* m *divisor de* q*, entonces existe un polinomio no trivial* $f \in \mathbb{F}_q[x]$ *de grado* m *tal que la aplicación* $f : \mathbb{F}_q \to \mathbb{F}_q$ *es un homomorfismo aditivo.*

Demostración: Por el Lema 10, podemos tomar $R = \mathbb{F}_q$ y un subgrupo $G \leq R$ de m elementos, $G = \{g_1, \ldots, g_m\}$,

$$\begin{array}{l}(0\ ,\alpha^{13},\alpha^{9}\ ,\alpha^{13},\alpha^{3}\ ,\alpha^{16},\alpha\ \ ,\alpha^{3}\ ,\alpha^{22},\alpha^{13},\alpha\ \ ,\alpha\ \ ,\alpha^{9}\ ,0\ \ ,1\ \ ,\alpha^{22},1\ \ ,\alpha^{16},\alpha^{3}\ ,\alpha^{14},\alpha^{16},\alpha^{9}\ ,1\ \ ,\alpha^{14},\alpha^{14},\alpha^{22})\\(\alpha^{14},\alpha^{22},1\ \ ,\alpha^{22},\alpha^{13},\alpha^{9}\ ,0\ \ ,\alpha^{13},\alpha^{3}\ ,\alpha^{22},0\ \ ,0\ \ ,1\ \ ,\alpha^{14},\alpha^{16},\alpha^{3}\ ,\alpha^{16},\alpha^{9}\ ,\alpha^{13},\alpha\ \ ,\alpha^{9}\ ,1\ \ ,\alpha^{16},\alpha\ \ ,\alpha\ \ ,\alpha^{3}\ )\\(\alpha\ \ ,\alpha^{3}\ ,\alpha^{16},\alpha^{3}\ ,\alpha^{22},1\ \ ,\alpha^{14},\alpha^{22},\alpha^{13},\alpha^{3}\ ,\alpha^{14},\alpha^{14},\alpha^{16},\alpha\ \ ,\alpha^{9}\ ,\alpha^{13},\alpha^{9}\ ,1\ \ ,\alpha^{22},0\ \ ,1\ \ ,\alpha^{16},\alpha^{9}\ ,0\ \ ,0\ \ ,\alpha^{13})\end{array} \tag{3}$$

$$\begin{array}{l}(\underline{\alpha}\ \ ,\alpha\ \ ,\alpha\ \ ,\alpha\ \ ,\alpha\ \ ,\underline{\alpha}\ \ ,\alpha\ \ ,\alpha\ \ ,\underline{\alpha}\ \ ,\alpha\ \ ,\underline{\alpha}\ \ ,\alpha\ \ ,\underline{\alpha}\ \ ,\alpha\ \ ,\alpha\ \ ,\alpha\ \ ,\underline{\alpha}\ \ ,\alpha\ \ ,\alpha\ \ ,\underline{\alpha}\ \ ,\underline{\alpha}\ \ ,\alpha\ \ )\\(\alpha^{3}\ ,\underline{\alpha^{3}}\ ,\alpha^{3}\ ,\underline{\alpha^{3}}\ ,\underline{\alpha^{3}}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\underline{\alpha^{3}}\ ,\underline{\alpha^{3}}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\underline{\alpha^{3}}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\underline{\alpha^{3}}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\underline{\alpha^{3}}\ )\\(\alpha^{9}\ ,\alpha^{9}\ ,\underline{\alpha^{9}}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\underline{\alpha^{9}}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\underline{\alpha^{9}}\ ,\alpha^{9}\ ,\underline{\alpha^{9}}\ ,\alpha^{9}\ ,\underline{\alpha^{9}}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\underline{\alpha^{9}}\ ,\underline{\alpha^{9}}\ ,\underline{\alpha^{9}}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\alpha^{9}\ )\end{array} \tag{4}$$

y construir un polinomio que tenga los elementos de G como raíces de multiplicidad 1,

$$f(x) = \rho \prod_{i=1}^{m} (x - g_i), \quad \rho \in \mathbb{F}_q \setminus \{0\}.$$

Nótese que, así como el polinomio $f(x)$ se anula cuando se evalúa en cualquier elemento del subgrupo G, el polinomio $f_\beta(x) = f(x) - f(\beta)$ se anula cuando se evalúa en la clase lateral $\beta + G$. Esto sucede porque $f(x)$ evalúa al mismo valor para cualquier elemento de $\beta + G$. Además,

$$\begin{aligned} f(-x) &= \rho \prod_{i=1}^{m} (-x - g_i) = \rho \prod_{i=1}^{m} (-x + g_i) \\ &= (-1)^m \rho \prod_{i=1}^{m} (x - g_i) = -\rho \prod_{i=1}^{m} (x - g_i) = -f(x). \end{aligned}$$

La penúltima igualdad se cumple para m impar. Este es el caso de cualquier cuerpo de característica diferente de 2. Para cuerpos de característica 2 tenemos que $-1 = 1$ y la igualdad también se cumple. Por último,

$$\begin{aligned} f(x+y) &= \rho \prod_{i=1}^{m} (x + y - g_i) = \rho \prod_{i=1}^{m} (x - (-y + g_i)) \\ &= \rho \prod_{i=1}^{m} (x - (-y - g_i)) = f_{-y}(x) = f(x) + f(y) \end{aligned}$$

demuestra que $f : \mathbb{F}_q \to \mathbb{F}_q$ es un homomorfismo aditivo. Nótese que $\ker f = G$ y $|\operatorname{im} f| = |\mathbb{F}_q / G|$. ■

Con la ayuda de estos resultados, podemos presentar la demostración del resultado principal del artículo.

Demostración del Teorema 8: Vamos a realizar la demostración construyendo, de nuevo, una pareja de subconjuntos del código que forma una (c, c)-configuración no separada.

Si $c^2 > q$, el código no es (c, c)-SFP por la Proposición 9. Entonces, asumiremos de aquí en adelante que $c^2 \leq q$. Bajo esta circunstancia, si c divide a q, c^2 también divide a q y por lo tanto q/c^2 es un valor entero. Como $k - 1 = n - d$, porque el código es MDS, tenemos que $k - 1 \geq q/c^2 + 1$. Considerando solamente el caso más restrictivo, $k = q/c^2 + 1$, quedará demostrado el resto de casos.

Por el Lema 11 podemos tomar un polinomio $f(x)$ no trivial de grado q/c^2 tal que $f : \mathbb{F}_q \to \mathbb{F}_q$ es un homomorfismo aditivo. Además, sabemos que $|\operatorname{im} f| = c^2$. Tomemos ahora un subgrupo $S \leq \operatorname{im} f$ de c elementos. Esto es posible porque el Lema 10 garantiza la existencia de dicho S. Consideremos también las c clases laterales de S, $\beta_1 + S, \ldots, \beta_c + S$, y los polinomios

$$f_j(x) = f(x) - \beta_j \quad 1 \leq j \leq c.$$

Obsérvese que estos polinomios evalúan a algún elemento del subgrupo S cuando $f_j(x) - \beta_j \in S$, es decir, cuando $f_j(x)$ toma el valor de un elemento de la clase $\beta_j + S$. Como las clases laterales de S son una partición de $\operatorname{im} f$, los c polinomios $f_j(x)$ replican el subgrupo S en posiciones disjuntas de $\operatorname{im} f$. Es rutinario comprobar que el mismo argumento justifica la replicación de cualquier clase lateral $\beta_r + S$, $1 \leq r \leq c$. De esta forma, el conjunto de las palabras generadas por dichos polinomios $T_2 = \{\mathbf{t}^j, 1 \leq j \leq c\}$, replican c conjuntos de c valores diferentes en c conjuntos de coordenadas disjuntas. Por último, tomamos el conjunto T_1 de las c palabras con valores constantes en una clase lateral arbitraria de S. Para esa clase lateral en concreto y para cualquier coordenada existe alguna palabra en T_2 que toma un valor en dicha clase lateral. Entonces tenemos que T_1, T_2 es una (c, c)-configuración no separada. ■

Por lo tanto, la demostración del Teorema 8 demuestra la equivalencia de las propiedades (1) para los códigos de Reed-Solomon cuando el tamaño máximo de la coalición es un divisor del tamaño del cuerpo.

IV. Ejemplo

Tomemos el cuerpo $\mathbb{F}_{27} = \mathbb{F}_3[x]/(x^3+2x+1)$ con elemento primitivo $\alpha = \overline{x}$. Consideremos que el tamaño máximo de las coaliciones es $c = 3$ y tomemos el código de Reed-Solomon $\mathrm{RS}(n = 26, k = 4)$. Tomemos ahora un subgrupo de $\mathbb{F}_q$ de tamaño $q/c^2 = 3$, $\{0, 1, \alpha^{13}\}$, y construyamos el polinomio

$$f(x) = (x - 0)(x - 1)(x - \alpha^{13}) = \rho(x^3 + \alpha^{13} x).$$

Por el Lema 11, la aplicación asociada al polinomio $f(x)$ es un homomorfismo. La palabra generada a partir de él es

$$\begin{aligned} &(0, \alpha^{13}, \alpha^9, \alpha^{13}, \alpha^3, \alpha^{16}, \alpha, \alpha^3, \alpha^{22}, \alpha^{13}, \alpha, \alpha, \\ &\quad \alpha^9, 0, 1, \alpha^{22}, 1, \alpha^{16}, \alpha^3, \alpha^{14}, \alpha^{16}, \alpha^9, 1, \alpha^{14}, \alpha^{14}, \alpha^{22}), \end{aligned}$$

de donde es inmediato ver que $\operatorname{im} f = \{0,\ 1,\ \alpha,\ \alpha^3,\ \alpha^9,\ \alpha^{13},\ \alpha^{14},\ \alpha^{16},\ \alpha^{22}\}$. Dado que $c^2 = |\operatorname{im} f|$, tomamos, por ejemplo, el subgrupo $S = \{0, 1, \alpha^{13}\} \leq \operatorname{im} f$ de c elementos y sus c clases laterales:

$$\begin{aligned} \beta_1 + S &= \{0, 1, \alpha^{13}\} \\ \beta_2 + S &= \{\alpha, \alpha^3, \alpha^9\} \\ \beta_3 + S &= \{\alpha^{14}, \alpha^{16}, \alpha^{22}\}, \end{aligned}$$

TABLA I
ALGUNAS FAMILIAS DE CÓDIGOS $\mathrm{RS}(n = q-1, k = \lceil n/c^2 + 1 \rceil)$:

(a) códigos que satisfacen $c^2 > q$; (b) códigos que satisfacen $c^2 | q$; (c) códigos que satisfacen $(k-1)|(q-1)$

$\mathbb{F}_q$	64	81	125	128	243	256	512	625	729	1024	2187
$c=2$	(b)	(c)	(c)	(b)	–	(b)	(b)	(c)	(c)	(b)	–
3	(c)	(b)	–	–	(b)	–	–	–	(b)	–	(b)
4	(b)	(c)	–	(b)	–	(b)	(b)	(c)	–	(b)	–
5	(c)	(c)	(b)	–	–	–	–	(b)	–	–	–
8	(b)	(c)	(c)	(b)	–	(b)	(b)	–	–	(b)	–
9	(a)	(b)	(c)	–	(b)	–	(c)	(c)	(b)	–	(b)
10	(a)	(a)	(c)	–	–	(c)	–	–	(c)	(c)	–
11	(a)	(a)	(c)	–	(b)	(c)	–	(c)	(c)	–	–
14-15	(a)	(a)	(a)	(a)	(c)	–	–	(c)	(c)	–	–
16	(a)	(a)	(a)	(a)	(a)	(b)	(b)	(c)	–	(b)	–
17-18	(a)	(a)	(a)	(a)	(a)	(a)	–	(c)	–	–	–

$\mathbb{F}_q$	512	625	729	1024	2187
$c=19$	–	(c)	–	(c)	–
20-22	–	(c)	(c)	(c)	–
23-24	(a)	(c)	(c)	–	–
25	(a)	(b)	(c)	–	–
26	(a)	(a)	(c)	–	–
27	(a)	(a)	(b)	–	(b)
28-31	(a)	(a)	(a)	–	–
32	(a)	(a)	(a)	(b)	–
33	(a)	(a)	(a)	(a)	–
34-46	(a)	(a)	(a)	(a)	(c)
≥ 47	(a)	(a)	(a)	(a)	(a)

donde consideramos que $\beta_1 = 0$, $\beta_2 = \alpha$ y $\beta_3 = \alpha^{14}$. Ahora construimos los polinomios $f_j(x) = f(x) - \beta_j, 1 \leq j \leq c$. Las palabras generadas a partir de ellos se muestran en (3), donde cada clase lateral $\beta_j + S$ se ha coloreado de la misma manera. Se puede observar que las palabras código replican las clases laterales en posiciones disjuntas. Por lo tanto, pueden generar un descendiente común con cualquier conjunto de palabras que esté formado por palabras constantes en una de las clases laterales de S. Por ejemplo, podemos tomar la clase $\beta_2 + S$, ilustrada en (4). Como vemos, los conjuntos de palabras (3) y (4) pueden generar un descendiente común (subrayado en (4)), y por lo tanto es una $(3,3)$-configuración no separada. De forma similar podríamos construir conjuntos para las clases $\beta_1 + S$ y $\beta_3 + S$.

V. RESULTADOS PARA OTROS TAMAÑOS DE COALICIÓN

En [4] se presentó un resultado relacionado, demostrando la equivalencia de las propiedades de trazabilidad de para otras familias de códigos de Reed-Solomon. La idea consiste en reformular la condición de (c,c)-SFP algebraicamente como un sistema de ecuaciones.

Teorema 12 ([4])**.** *Sea* $\mathrm{RS}(n,k)$ *un código de Reed-Solomon sobre* $\mathbb{F}_q$ *tal que* $k-1$ *divide* $q-1$*. Entonces, si* $d \leq n - n/c^2$ *el código no es* (c,c)*-SFP.*

Esto cubre familias de códigos de Reed-Solomon con $c^2 \geq (q-1)/(k-1)$ siendo $(q-1)/(k-1)$ un entero.

Proposición 13. *Dado el código de Reed-Solomon* $\mathrm{RS}(n,k)$ *sobre* $\mathbb{F}_q$ *con distancia mínima* d *y un entero* c*, si* $c|q$ *ó* $(k-1)|(q-1)$ *entonces,* $\mathrm{RS}(n,k)$ *satisface*

$$d > (1 - 1/c^2)n \Leftrightarrow c\text{-}TA \Leftrightarrow c\text{-}IPP \Leftrightarrow (c,c)\text{-}SFP.$$

Ilustrativamente, en la Tabla I mostramos algunas familias de códigos de Reed-Solomon, para determinados valores de c y q, tales que satisfacen (c,c)-SFP $\Leftrightarrow$ c-TA con $k = \lceil (q-1)/c^2 + 1 \rceil$. Esto sugiere una respuesta afirmativa a la pregunta planteada en [1], [2].

VI. CONCLUSIÓN

En este artículo hemos estudiado las propiedades de trazabilidad de los códigos de Reed-Solomon. Nuestro principal objetivo era el de dar una respuesta a la pregunta planteada por Silverberg et al. en [1], [2]: *¿Es cierto que todos los códigos Reed-Solomon IPP son también TA?*. Hemos demostrado la equivalencia de las propiedades SFP, IPP y TA para algunas familias de códigos de Reed-Solomon, cuando el tamaño de la coalición de traidores tiene la particularidad que su tamaño máximo divide el tamaño del cuerpo. Obviamente, esto no responde al cien por cien la pregunta planteada, pero esperamos que contribuya a aportar ideas que puedan ser útiles para encontrar la respuesta completa.

AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente por el Proyecto CICYT TEC2008-06663-C03-01 (P2PSec), por el Ministerio de Educación y Ciencia mediante el Proyecto CONSOLIDER CSD2007-00004 (ARES) y por la Generalitat de Catalunya mediante la ayuda AGAUR SGR 1362.

REFERENCIAS

[1] A. Silverberg, J. Staddon, and J. Walker, "Efficient traitor tracing algorithms using list decoding," in *In Proceedings of ASIACRYPT '01, volume 2248 of LNCS*, 2001, pp. 175–192.

[2] A. Silverberg, J. Staddon, and J. L. Walker, "Applications of list decoding to tracing traitors," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1312–1318, May 2003.

[3] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1897–1905, 1998.

[4] M. Fernandez, J. Cotrina, M. Soriano, and N. Domingo, "A note about the identifier parent property in Reed-Solomon codes," *Computers & Security*, 2010, DOI: 10.1016/j.cose.2009.12.012.

[5] J. N. Staddon, D. R. Stinson, and R. Wei, "Combinatorial properties of frameproof and traceability codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 1042–1049, 2001.

[6] H. Jin and M. Blaum, "Combinatorial properties for traceability codes using error correcting codes," *IEEE Trans. Inf. Theory*, vol. 53, no. 2, pp. 804–808, 2007.

[7] G. D. Cohen and H. G. Schaathun, "Upper bounds on separating codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1291–1294, 2004.

[8] J. J. Rotman, *An Introduction to the Theory of Groups*. Springer-Verlag New York, Inc., 1995.

Estudio sobre el uso de códigos LDPC en esquemas de fingerprinting.

Sergi Vendrell
Departamento de Ingeniería Telemática.
Universitat Politècnica de Catalunya.
Email:sergi.stanfm@gmail.com

Joan Tomàs-Buliart
Departamento de Ingeniería Telemática.
Universitat Politècnica de Catalunya.
Email:jtomas@entel.upc.edu

Marcel Fernandez
Departamento de Ingeniería Telemática.
Universitat Politècnica de Catalunya.
Email:marcel@entel.upc.edu

Miguel Soriano
Departamento de Ingeniería Telemática.
Universitat Politècnica de Catalunya.
CTTC: Centre Tecnològic de Telecomunicacions de Catalunya.
Parc Mediterrani de la Tecnologia (PMT).
Email:soriano@entel.upc.edu

***Resumen*—El presente artículo presenta un estudio sobre la utilización de los códigos LDPC en aplicaciones de fingerprinting. Concretamente su utilización en entornos que sean susceptibles de ataques de confabulación. Aunque esta aplicación de los LDPC es nueva, los resultados aquí presentados demuestran que puede ser una buena solución para abordar este problema en entornos dónde se necesitan pocos usuarios, del orden de decenas de miles, y en los que el contenedor del fingerprint tiene una capacidad más bien pequeña. En este sentido se han conseguido probabilidades de éxito superiores al 98 % con 28 confabuladores de entre 1024 usuarios.**

I. Introducción

La protección de los derechos de autor o copyright se ha convertido en un tema importante dentro de la comunidad científica durante los últimos años. Aunque en primer término se abogo por sistemas que impidieran la replicación de los soportes, pronto se llego a la conclusión que estos sistemas, una vez rotos, nada podía parar la replicación ilícita. Durante la última década ha ido asentándose la idea de que quizás sería más apropiado enfocar el problema desde una óptica más proactiva: el fingerprinting. El concepto de fingerprinting fue introducido por Wagner en [1] como un método para proteger la propiedad intelectual de contenidos multimedia. La idea básica es generar copias del mismo contenido pero que, de alguna manera, puedan identificar su comprador. La finalidad del fingerprinting no es evitar las copias ilícitas sino disuadir a los usuarios de hacerlas.

El principal ataque a los esquemas de fingerprinting son los ataques de confabulación, es decir, cuando dos o más usuarios se unen para comparar sus copias, encontrar diferencias y generar una nueva copia con partes de las tres que pueda ocultar sus identidades para evitar ser inculpados. El problema se agrava cuando no sólo no se identifica ninguno de los traidores sino que, además, se inculpa un usuario inocente. Por lo tanto, el objetivo del fingerprinting es encontrar un sistema de marcas que ayuden a prevenir los ataques de confabulación.

Boneh y Shaw presentaron en [2] una de las primeras propuestas sobre códigos seguros frente a confabulaciones. Su propuesta es capaz de identificar a un usuario ilícito entre una coalición de c usuarios con una probabilidad de error ϵ. Su construcción se basa en el uso de un código interno binario y un código aleatorio como código externo.

El rendimiento de los códigos LDPC conjuntamente con uno de sus algoritmos de decodificación soft como el algoritmo Sum-Product (SPA) como códigos seguros frente a confabulaciones será analizado en profundidad teniendo en cuenta su comportamiento en función de la longitud del código y el número de usuarios que forman la coalición. La capacidad del decodificador LDPC para identificar al menos uno de los traidores y el número de falsos positivos generados será de ayuda para discutir la eficacia de estos códigos en según que escenarios. En otros esquemas fingerprinting [3] se propone el uso de códigos LDPC para mejorar el rendimiento de los BIBD frente a ruido gaussiano, la idea en nuestra propuesta es utilizarlos directamente como código fingerprinting y no solamente para combatir el efecto del ruido en el canal.

Este artículo esta organizado de la forma siguiente. La sección II presenta algunos conceptos y definiciones referentes al fingerprinting y a los códigos correctores de error. La sección III introduce los códigos Low-Density Parity-Check y el algoritmo de decodificación Sum-Product (SPA). Las contribuciones de esta propuesta son expuestas en la sección IV mientras que los resultados que de ellas se han derivado se presentan en la sección V. Finalmente, la sección VI resume las conclusiones de este artículo.

II. Esquemas de fingerprinting

El fingerprinting es una técnica que consiste en incrustar alguna característica única en las copias de un contenido digital de forma que permita identificar una de estas copias de forma unívoca.

La protección del copyright frente a redistribuciones de contenido digital es uno de los objetivos principales del fingerprinting. Cuando una organización trata de distribuir contenido digital protegido, un fingerprint distinto es incrustado en cada una de las copias enviadas. El objetivo es que estos fingerprints se puedan usar como evidencia irrefutable para identificar el receptor o comprador de una copia concreta de este contenido. Si se detectan intenciones maliciosas, la extracción del fingerprint de la copia en conflicto debe ser prueba suficiente para inculpar al usuario fraudulento.

Como los bits que forman el fingerprinting son distribuidos dentro del documento siguiendo algún patrón, para facilitar su posterior extracción, es de esperar la colaboración entre varios usuarios para conseguir realizar un ataque de confabulación. Nada les impide comparar sus copias y generar una nueva copia pirata que intente ocultar su identidad con el fin de no poder ser inculpados. Por lo tanto, uno de los requisitos más importantes exigidos a los códigos fingerprinting es que sean resistentes a confabulaciones de un cierto número de usuarios.

En [2], [4], [5], [6] se discuten en profundidad las características necesarias para obtener un esquema de fingerprinting fiable. Estas características se pueden resumir en cuatro: mínima probabilidad de error o falso positivo, cardinalidad del código sustancial, longitud de código mínima y facilidad de rastreo. Como parece lógico, un código capaz de cumplir todas estás características puede ser imposible de encontrar ya que, en cierto modo, entran en contradicción. Por lo tanto, es primordial conocer los requisitos de la aplicación en concreto para cuantificar la importancia o relevancia de cada una de ellas.

A continuación se presentan algunas definiciones útiles para centrar los objetivos de este artículo. Básicamente las dividimos en dos grupos, las relativas a la *Marking Assumption* y las relativas a los *códigos n-seguros*.

II-A. Marking Assumption

La *Marking Assumption* se puede considerar el principio básico en el diseño de los códigos fingerprinting. De hecho establece las reglas básicas relativas a los ataques de confabulación. Antes de definirla son necesarias unas definiciones previas.

Dada w, una palabra código de longitud n, tal que $w \in \Sigma$ y un conjunto $I = (i_1, i_2, \cdots, i_r)$ donde $1 \leq r \leq n$, entonces $w \mid_I$ es la palabra $(w_{i_1}, w_{i_2}, \cdots, w_{i_r})$ donde w_i es el $i-$ésimo elemento de w. $w \mid_I$ denota la restricción de w en las posiciones especificadas por I.

Definition 1: (**Código** en [2]) Un conjunto $\Gamma = \{w^{(1)}, w^{(2)}, \cdots, w^{(n)}\} \subseteq \Sigma^l$, donde Σ denota un alfabeto de tamaño s, será referido como un (l, n)-código. La palabra código $w^{(u_i)}$ será asignada a un usuario u_i, para $1 \leq i \leq n$. Nos referiremos al conjunto de palabras de Γ como **código**.

Definition 2: (**Posiciones indetectable** en [2]) Sea $\Gamma = \{w^{(1)}, w^{(2)}, \cdots, w^{(n)}\}$ un (l,n)-código y $C = \{u_1, u_2, \cdots, u_c\}$ una coalición de c-traidores. La posición i será **indetectable** para C si las palabras asignadas a los usuarios en C coinciden en la posición i-ésima, es decir $w_i^{(u_1)} = \cdots = w_i^{(u_c)}$.

El concepto más importante recae en la definición de *feasible set*. Aunque se han publicado varias definiciones posibles para este concepto, la definición de Bonew-Shaw es la siguiente.

Definition 3: (**Feasible Set** en [2]) Sea $\Gamma = \{w^{(1)}, w^{(2)}, \cdots, w^{(n)}\}$ un (l,n)-código y $C = \{u_1, u_2, \cdots, u_c\}$ una coalición de c-traidores. Se define **feasible set** Γ de C como

$$\Gamma(C) = \{x = (x_1, \cdots, x_l) \in \Sigma^l \mid x_j \in w_j, 1 \leq j \leq l\}$$

donde

$$w_j = \begin{cases} \{w_j^{(u_1)}\} & w_j^{(u_1)} = \cdots = w_j^{(u_c)} \\ \{w_j^{(u_i)} \mid 1 \leq i \leq c\} \cup \{?\} & otherwise \end{cases}$$

donde ? denota una posición borrada.

Definition 4: (Marking Assumption) Sea $\Gamma = \{w^{(1)}, w^{(2)}, \cdots, w^{(n)}\}$ un (l,n)-código, $C = \{u_1, u_2, \cdots, u_c\}$ una coalición de c-traidores y $\Gamma(C)$ el feasible set de C. La coalición C es solo capaz de crear un objeto cuyo fingerprint se encuentre dentro de $\Gamma(C)$.

Según esta definición, cualquier coalición sólo es capaz de detectar las posiciones donde los fingerprints difieren entre un usuario y otro. Por lo tanto, los usuarios maliciosos pueden escoger entre dejar uno de sus valores o bien introducir un borrado en una determinada posición.

III. Códigos Low-Density Parity-Check

Los códigos Low-Density Parity-Check o LDPC, fueron propuestos en primer lugar por Robert G. Gallager [7], [8] en 1962. Desafortunadamente, el trabajo de Gallager fue ignorado durante muchos años debido a la falta de recursos computacionales de la época. No fue hasta finales de los años 90 cuando Mackay y Neal [9] empezaron a investigar códigos basados en grafos y decodificación iterativa. Hoy en día los LDPC gozan del reconocimiento que se merecen gracias a su elevado rendimiento que los sitúa muy cerca del límite de Shannon. De hecho, actualmente se consideran unos duros competidores con los Turbo códigos en cuanto a corrección de errores se refiere en entornos donde se necesita una elevada fiabilidad. Además los LDPC presentan importantes ventajas

respecto de los Turbo códigos ya que no requieren de un sofisticado entrelazador para ofrecer un rendimiento óptimo, su rendimiento en cuanto a error de bloque es superior, el suelo de error está situado en niveles de BER muy inferiores y su decodificación no se basa en trellis por lo que ofrecen una velocidad de decodificación mayor.

III-A. Códigos Parity-Check

Un código *parity-check* de longitud N es un código bloque binario y lineal donde todas sus palabras código cumplen M restricciones lineales en cuanto a validación de paridades. El código se define mediante una matriz de paridad H de dimensiones $M \times N$, cada fila de la cual especifica una determinada de las M restricciones. Por lo tanto el código *parity-check* es el conjunto de vectores c que satisfacen todas las M restricciones, es decir, $c \cdot H^T = 0$. La característica diferencial de un LDPC es que en este caso, la matriz de paridad siempre esta definida de forma que sea poco densa, es decir, que contenga pocos 1.

Definition 5: La matriz de un código (j,k) LDPC regular es una matriz binaria de dimensiones $M \times N$ que tiene exactamente j unos en cada columna y exactamente k unos en cada fila, donde $j < k << N$.

A partir de la definición anterior se puede deducir que cada ecuación de validación de paridad afecta a k bits, y cada uno de ellos afecta a exactamente j ecuaciones de validación de paridad. Para asegurar que el ratio del código no es nulo, la restricción $j < k$ es necesaria para evitar precisamente que el vector nulo pueda satisfacer todas las restricciones.

III-B. Algoritmos de decodificación

Los códigos LDPC pueden descodificarse utilizando tanto algoritmos de hard-decision o de soft-decision. Como regla general, los algoritmos de hard-decision, como por ejemplo el Majority-logic (MLG) o el Bit-flipping (BF), requieren menos complejidad de decodificación y suelen ser más rápidos. Sin embargo, su rendimiento es inferior en comparación con los algoritmos de soft-decision. Por otro lado, existen algoritmos basados en soft-decision como el de probabilidad a posteriori (APP) y el de decodificación iterativa basado en propagación de confianza (IDBP), también conocido como Sum-product (SPA), que, aunque penalizan la velocidad de decodificación, ofrecen un rendimiento sustancialmente superior. En este estudio se ha optado por el SPA ya que es el que ofrece un rendimiento mayor en cuanto a corrección de errores. A continuación vamos a definir algunos aspectos referentes al entorno de aplicación.

Sobre un canal con ruido gausiano aditivo y blanco (AWGN) de media nula y PSD de N_0, se usa un código LDPC para control de errores. Una palabra código $v = (v_0, v_1, \ldots, v_{n-1})$ es codificada como la secuencia bipolar $x = (x_0, x_1, \ldots, x_{n-1})$ donde:

$$x_l = \begin{cases} +1 & \text{si } v_l = 1 \\ -1 & \text{si } v_l = 0 \end{cases}$$

Sea $y = (y_0, y_1, \ldots, y_{n-1})$ una secuencia real recibida con $y_l = \pm 1 + n_l$, siendo n_l una variable Gaussiana aleatoria de media cero y varianza $\frac{N_0}{2}$. En este caso, la secuencia discreta $z = (z_0, z_1, \ldots, z_{n-1})$ se obtiene a partir de y mediante:

$$z_l = \begin{cases} 1 & \text{si } y_l > 0 \\ 0 & \text{si } y_l \leq 0 \end{cases}$$

Sea $h_1, h_2, \ldots, h_J$ la forma de denotar las filas de la matriz de paridad H, donde $h_i = (h_{i,0}, h_{i,1}, \ldots, h_{j,n-1})$ para $1 \leq i \leq J$, entonces,

$$s = (s_1, s_2, \cdots, s_J) = z \cdot H^T \tag{1}$$

genera el síndrome de la secuencia recibida z, donde el i-ésimo componente s_i, es dado a partir de la siguiente ecuación

$$s_i = z \cdot h_i = \sum_{l=0}^{n-1} z_l h_{i,l} \tag{2}$$

El vector recibido z será una palabra código si y solo si $s = 0$. En caso contrario, significa que la secuencia z contiene los errores indicados por las ecuaciones de paridad no satisfechas. El número de fallos en la comprobación de la paridad es igual al número de componentes distintos de cero presentes en s.

El algoritmo SPA procesa los símbolos recibidos de forma iterativa para mejorar la fiabilidad de cada uno de ellos basándose en la computación de las validaciones de paridad a partir de los símbolos discretos y la matriz H. La fiabilidad del símbolo es medida mediante su ratio log-likelihood (LLR). Esta fiabilidad medida al final de cada iteración es usada como entrada para la siguiente iteración. El proceso de decodificación continua hasta que se cumple un determinado criterio.

La implementación del algoritmo SPA recae en la computación de las probabilidades a posteriores marginales, $P(v_l|y)$, para $0 \leq l < n$ donde y es la secuencia real recibida. El LLR para cada bit se define como

$$L(v_l) = log \frac{P(v_l = 1|y)}{P(v_l = 0|y)}. \tag{3}$$

Existen varias versiones del algoritmo SPA. En las simulaciones presentadas en este artículo se usa el esquema presentado por Radford M. Neal y Bagawan [10], [11]. La mejora aportada en cuanto a la información extrínseca en cada iteración es usada para mejorar tanto la información extrínseca de salida como la fiabilidad de cada símbolo en la próxima iteración. El proceso continua hasta que se consigue encontrar una secuencia que satisfaga todas las restricciones impuestas por el código, es decir, la matriz H.

IV. Propuesta

El principal objetivo de esta propuesta radica en el análisis del rendimiento de los LDPC en escenarios donde aparecen ataques de confabulación. Estos entornos requieren esquemas con una importante fiabilidad y una elevada capacidad de encontrar como mínimo uno de los usuarios que han formado parte de la coalición, con el requisito ineludible de prevenir la inculpación de usuarios inocentes.

El esquema propuesto consiste en la concatenación de códigos LDPC con un código de repetición para crear un fingerprint capaz de identificar cada usuario. En un primer paso, se generan Q cadenas binarias aleatorias de longitud M bits, cada una de ellas será el identificador u_i del usuario i donde $1 \leq i \leq Q$. Todos estos identificadores u_i son codificados mediante un código LDPC con un ratio de $N/M = 2$, generando Q palabras código de longitud $N = 2M$. Cada bit de estas palabras código se codifican mediante un código de repetición, la longitud del cual se fijará en función de probabilidad de detección de colusión deseada en una determinada posición. En lo referente a la generación del código LDPC, se utiliza un código regular con una matriz de 3 unos por columna evitando ciclos de Tanner de grado 4.

En nuestras simulaciones consideramos que cualquier subgrupo de c usuarios entre los Q usuarios pueden realizar un ataque de confabulación. Una vez la confabulación ha tenido lugar, suponemos que una cierta cantidad de ruido de canal puede ser añadido al objeto resultante de la confabulación. Finalmente, el resultado se descodifica mediante el algoritmo SPA y se aplica un *mached filter* basado en correlación.

Un tema importante a tener en cuenta es el modelado de la confabulación. En este sentido, en las simulaciones se han considerado varias alternativas. Básicamente, cuando los confabuladores comparan sus copias advierten que en determinadas posiciones hay diferencias. Teniendo en cuenta la *Marking Assumption*, ante este hecho los confabuladores pueden actuar de distintas maneras. Cada manera o forma de actuar marcará un tipo distinto de confabulación:

- **Erasure o Borrado:** En este caso deciden poner un valor no válido en esta posición.
- **Majority o Mayoría:** En este caso deciden poner el valor más repetido entre sus copias.
- **Average o Media:** El valor en esta posición refiere al valor de la media de los valores de todos los usuarios en esta posición.
- **Coin-flipping o Tiro de moneda:** Deciden poner uno de los valores detectados de forma equiprobable entre todos los valores detectados. En el caso binario sería con probabilidad $1/2$ cada valor.

El decodificador LDPC implementado permite dos tipos distintos de salida: una palabra soft, es decir, la verosimilitud de los bits descodificados, y su correspondiente palabra hard. A partir de la salida del descodificador LDPC, se correlan los distintos identificadores de usuario con la salida soft o hard en función del esquema elegido. Finalmente se considera que el usuario con una correlación mayor es miembro de la confabulación y, por lo tanto, es inculpado.

Con el objetivo de extraer conclusiones respeto al rendimiento de los LDPC como medida de protección frente a los ataques de confabulación, los resultados de las simulaciones se muestran en función de la probabilidad de identificación de traidores (TIP), es decir, cuantas veces el usuario inculpado ha participado realmente en el ataque de confabulación.

En este artículo, el rendimiento de los LDPC en entornos con ataques de confabulación se ha evaluado en función del numero de confabuladores que pueden formar parte de la coalición, exigiendo un rendimiento mínimo de un TIP del 98 %. Cabe destacar que se han usado códigos con una longitud extremadamente corta (128 y 256 bits) para las longitudes habituales en códigos fingerprinting ya que uno de los requisitos exigidos a nuestro sistema es un buen rendimiento con longitudes cortas ya que existen varios códigos fingerprinting con un buen rendimiento con longitudes grandes.

V. Resultados de las simulaciones

El uso de códigos LDPC conjuntamente con el algoritmo de decodificación SPA usando propagación de confianza (IDBP) con 50 iteraciones es una potente herramienta gracias al vector de verosimilitud proporcionado por el decodificador. La clave reside en analizar los bits a la entrada intentando discernir cuales han sido detectados por los confabuladores y cuales no. Ayudar en esta tarea es el objetivo del código de repetición, básicamente, cuanto más largo sea el código de repetición, más fiable será el valor que utilizará el LDPC para descodificar. Una vez hecha esta estimación se asigna una mayor probabilidad a la entrada del descodificador a los bits que se considera que no han sido detectados. Como el SPA es el algoritmo escogido para todas nuestras simulaciones, esta asignación se comporta de la misma forma tanto en los casos de salidas hard como en salidas soft.

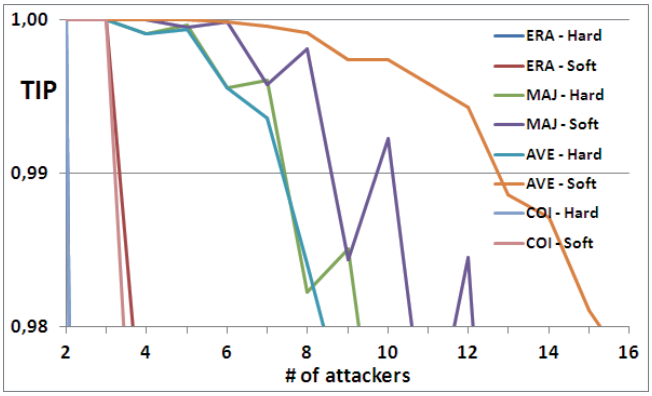
Considerar que el usuario que tiene una mayor correlación con la salida del descodificador es inculpado, implica la posibilidad de inculpar un inocente si la correlación entre su identificador y la palabra descodificada es la mayor. Esto sucede cuando se produce un error en la decodificación y recibe el nombre de falso positivo. Los falsos positivos deben de evitarse o, como mínimo minimizarlos todo lo posible.

V-A. Rendimiento del descodificador LDPC vs. número de confabuladores

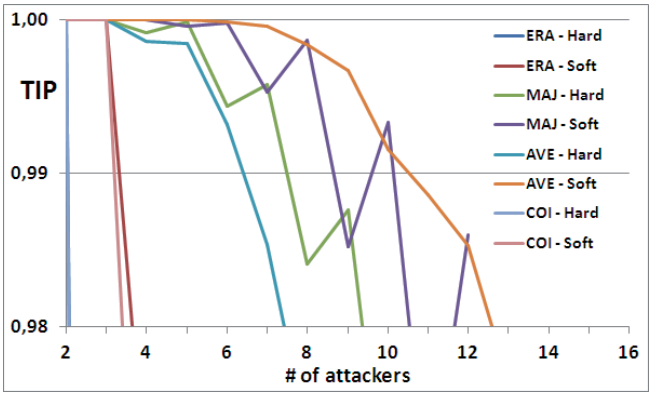
El número de usuarios que forman parte de la coalición guarda una relación directa con el rendimiento del sistema

y es un valor que el descodificador no conoce, es decir, a partir del fingerprint recuperado no hay forma de saber cuantos traidores han participado en la confabulación. Por lo tanto, es importante saber cuantos traidores puede soportar el decodificador al mismo tiempo sin que esto afecte gravemente su rendimiento, es decir, si que el TIP sea inferior al 98 %. En las simulaciones se ha considerado que el número de usuarios del sistema era 1024 (es decir $Q = 1024$).

Las simulaciones realizadas utilizan un identificador de 128 o 256 bits (es decir, $M = 128$ o $M = 256$) y, como el codificador LDPC escogido presenta una relación de $M/N = 1/2$, la salida, N, será de 256 o 512 bits respectivamente. El limite establecido para los falsos positivos es el 2 % por lo tanto, los valores en que el TIP sea inferior del 98 % no se han considerado como válidos. Por otro lado, se ha diferenciado el caso en los que se consideraba un canal sin ruido y el caso con un canal con ruido AWGN.



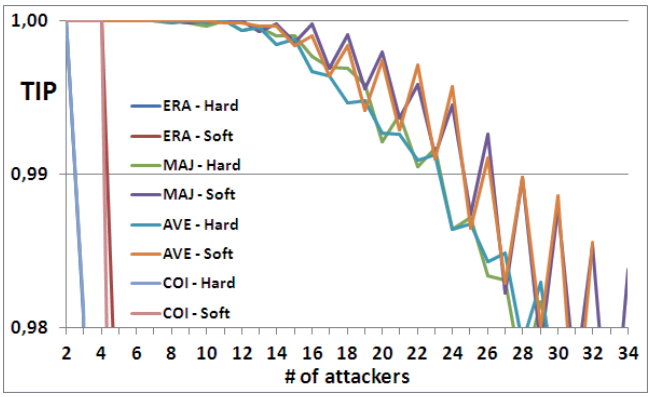
(a) $M = 128$ bits sin AWGN



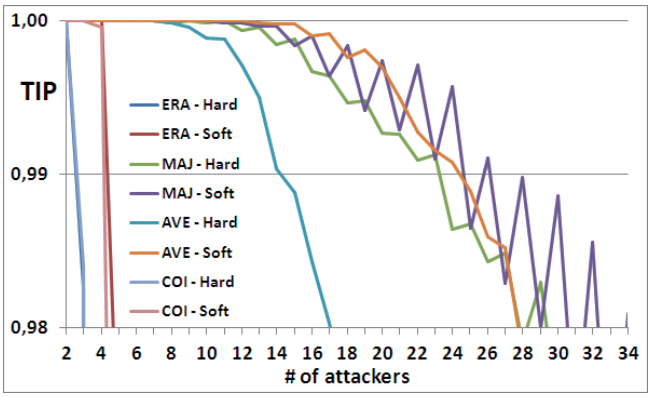
(b) $M = 128$ bits con AWGN

Figura 1. Resultados para M=128 bits

Los valores del TIP en función de M, el método de confabulación, la presencia o ausencia de ruido en el canal y el tipo de decodificación (hard o soft) se muestran en las figuras 1 y 2. Como podemos observar, los métodos de ataque por mayoría y media son los que ofrecen un mejor rendimiento des del punto de vista de la decodificación. Este resultado es mejorado cuando el descodificador utiliza la salida soft en lugar de la hard.



(a) $M = 256$ bits sin AWGN



(b) $M = 256$ bits con AWGN

Figura 2. Resultados para M=256 bits

Podemos observar que cuando el número de traidores aumenta, el rendimiento empeora hasta el punto en el que el TIP disminuye por debajo del 98 %, momento en el que se considera que el sistema no es viable. Por otro lado, observamos que la presencia de ruido no afecta todos los sistemas de confabulación del mismo modo. Mientras que cuando la confabulación escogida por los atacantes es la media, el ruido provoca una caída de rendimiento, en cambio, la presencia de ruido apenas afecta cuando la confabulación se realiza mediante el método de mayoría.

En cuanto a la capacidad del sistema, teniendo en cuenta que nuestro sistema esta dimensionado para 1024 usuarios, podemos comprobar que, si des de la capa de watermarking se puede forzar que los únicos ataques de confabulación posibles sean el de media o el de mayoría, y el sistema de decodificación es soft, podemos llegar a identificar, aún con la presencia de ruido AWGN, un traidor de un grupo de 28 con una probabilidad de éxito del 98 % con tan solo 512 bits de marca ($M = 256$). Cabe destacar que esto significa que más del 2,7 % de los usuarios del sistema participan en la confabulación, lo cual es un valor extremadamente grande.

VI. Conclusiones

El presente artículo discute los resultados obtenidos con la utilización de los códigos de LDPC como códigos fingerprinting para combatir ataques de confabulación. Se ha mostrado mediante simulaciones que la probabilidad de identificación de traidores (TIP) decrece cuando el número de usuarios que participan en la confabulación aumenta. Aún así, el algoritmo SPA presenta un rendimiento excelente con unas longitudes de código relativamente pequeñas aunque el número de atacantes sea sustancial.

En este sentido se han conseguido probabilidades de éxito superiores al 98 % con 28 confabuladores de entre 1024 usuarios. En cualquier caso, cabe destacar que el rendimiento crece de forma sustancial cuando la longitud de la marca aumenta. Por otro lado, aunque los resultados se han condicionado a distintos tipos de ataques de confabulación, estos ataques se pueden forzar en función del tipo de watermarking con el que se integre este esquema de fingerprinting.

Finalmente, aún teniendo en cuenta el estado embrionario de esta aplicación de los LDPC, los resultados preliminares aquí expuestos animan a seguir investigando en este camino.

Agradecimientos

Este trabajo ha sido financiado en parte por el Gobierno Español con el programa Ingenio 2010 con el proyecto ARES (Advanced Research on Information Security and Privacy – CONSOLIDER CSD2007-00004) y por la CICYT con el proyecto P2PSEC (Provision segura de servicios sobre P2P - TEC2008-06663-C03-01). Además de por la AGAUR de la Generalitat de Catalunya con el proyecto SGR – 1362.

Referencias

[1] Neal R. Wagner. Fingerprinting. In *SP '83: Proceedings of the 1983 IEEE Symposium on Security and Privacy*, page 18, Washington, DC, USA, 1983. IEEE Computer Society.

[2] Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data (extended abstract). In *CRYPTO '95: Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, pages 452–465, London, UK, 1995. Springer-Verlag.

[3] Jie Yang, Ping Liu, and GuoZhen Tan. The digital fingerprint coding based on ldpc. In *Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference on*, volume 3, pages 2600 – 2603 vol.3, 31 2004.

[4] Tina Lindkvist. Fingerprinting of digital documents. *Dissertation No 706*, 2001.

[5] Francesc Sebé and Josep Domingo-Ferrer. Short 3-secure fingerprinting codes for copyright protection. In *ACISP '02: Proceedings of the 7th Australian Conference on Information Security and Privacy*, pages 316–327, London, UK, 2002. Springer-Verlag.

[6] M. Fernández and Miguel Soriano. Fingerprinting concatenated codes with efficient identification. In *ISC '02: Proceedings of the 5th International Conference on Information Security*, pages 459–470, London, UK, 2002. Springer-Verlag.

[7] R.G.Gallager. Low density parity check codes. pages 21–28. IRE Trans.Inform. Theory, 1962.

[8] R.G.Gallager. Low density parity check codes. Cambridge, 1963. MIT Press.

[9] David J.C. MacKay and Radford M. Neal. Near shannon limit performance of low density parity check codes. *Electronics Letters*, 32:1645–1646, 1996.

[10] Radford M. Neal. Radford m. neal's home page. *http://www.cs.toronto.edu/ radford/*, 2000.

[11] Bagawan S. Nugroho. Digital communication systems simulation using matlab and c mex. *http://bsnugroho.googlepages.com/*, 2002.

Índice de autores