13

Matemáticas y Criptografía

Por
Alejandro Melle Hernández
Universidad Complutense de Madrid

Abstract

We live in an information-based society. The techniques to keep secret the information are the field of study of cryptography. The cryptography are the mathematical tools related to confidentiality, data integrity, entity authentication and data origin authentication.

There are two forms of encryption: Symmetric Key Encryption which is fast but both parties need to know a shared secret key and Public Key Encryption which is slow but only one party needs to keep a key private.

The first and most popular public key exchange algorithm is RSA. The security of RSA is based on the intractability of the integer factorization problem. Diffie-Hellman key exchange relies on difficulty of computing discrete logarithms. There are a few other key exchange schemes that are used in practice, for example, the Digital Signature Algorithm (DSA) and the Elliptic Curve Digital Signature Algorithm (ECDSA). The security of those schemes is based on the discrete logarithm problem in the multiplicative group of a prime field or in the group of points of an elliptic curve over a finite field.

One of the fundamental tools used in information security is the signature. From practical view point the main ingredient of the signature in cryptology are the hash functions. The main role of a cryptographic hash function is in the provision of digital signatures. Since hash functions are generally faster than digital signature algorithms, it is typical to compute the digital signature to some document by computing the signature on the document's hash value, which is small compared to the document itself. It was a big surprise that a vulnerability of standard hash functions was announced in Feb. 2005. The attack primarily affects some digital signature applications, including timestamping and certificate signing operations, where one party prepares a message for the generation of a digital signature by a second party, and third parties then verify the signature.

When quantum computers reach approximately 30 to 40 q-bits they will start to have the speed (parallelism) needed to attack the methods society uses to protect data and processes, including encryption, digital signatures, random number generators, key transmission, and other security algorithms. In particular all the standards used nowdays will became obsolete and one

should be prepare to have ready several other cryptosystems that are conjectured to resist quantum computers, how efficient are they, in theory and in practice?

To know something else: http://www.cryptomathic.com/labs/ellipticcurves.html

Second Cryptography Hash Workshop http://www.csrc.nist.gov/pki/HashWorkshop/index.html

Mathematics and Internet security. http://www.mathaware.org/mam/06/

Introducción

A nadie parece extrañar la frase "... hoy vivimos en la sociedad de la información..", cada día cualquiera de nosotros almacena, procesa e incluso transmite información en forma digital. Sin duda, desde los años ochenta, el uso de los ordenadores, de las redes locales, de Internet, del comercio electrónico, de la telefonía inalámbrica, del *wireless*... es parte de nuestra vida diaria.

La contrapartida al uso de toda esa información sobre redes abiertas son los problemas de *privacidad* y *seguridad*. Se deben pues buscar mecanismos tanto legales como científico-técnicos para garantizar ambas. Un ejemplo de plena actualidad es la puesta en marcha, en un futuro cercano, del Documento Nacional de Identidad electrónico.

La *Criptografía* es la parte científico-técnica que trata de garantizar los fundamentos de la seguridad en la transmisión de información. Los objetivos básicos que persigue la Criptografía son:

- *Confidencialidad*, se debe mantener la información en secreto para toda entidad excepto para los que tengan autorizado el acceso a ella.
- *Integridad de los datos*, se debe garantizar que el mensaje no ha sido modificado durante la transmisión.
- *Autenticidad*, el receptor del mensaje debe ser capaz de verificar quién es el emisor y, por otro lado, ambos deben ser capaces de identificarse mutuamente.
- *No repudio*, el emisor no debe ser capaz de negar que ha transmitido el mensaje.

Para mantener la confidencialidad de la información, el *texto en claro* (que puede ser texto, datos, ...) debe ser codificado antes de ser transmitido, obteniéndose el *texto cifrado*. Una vez enviado el texto cifrado el receptor debe *descifrar* el mensaje. La idea básica de *cifrar* el texto es garantizar que no haya una tercera parte no autorizada que sea capaz de acceder a la información sin conocer la clave.

El *Criptoanálisis* es la ciencia que estudia los ataques de agentes extraños contra los criptosistemas. El objetivo de los ataques es variable y depende de las habilidades del atacante. Puede ser que se pretenda recuperar texto en claro del texto cifrado, o bien que se pretenda substituir parte del texto en claro original, o bien falsificar firmas digitales... Uno de las principios del Criptoanálisis es el *principio de Kerkhoff* que establece que el criptosistema debe ser de conocimiento público, incluyendo los algoritmos y su implementación, y que lo único que se debe mantener en secreto es la clave con la que se cifra. Lo que distingue la criptografía moderna de la clásica es que en la criptografía moderna se sigue a rajatabla este principio.

En esta nota quiero presentar y discutir las componentes primitivas y los criptosistemas más usados, *primitivos* se refiere a que son las piezas básicas de otros criptosistemas.

Criptosistemas de clave simétrica

Un método que se viene usando desde la antigüedad para garantizar la confidencialidad es que tanto el emisor como el receptor se pongan de acuerdo previamente en una *clave* k que les permita tanto cifrar como descifrar sus mensajes, manteniendo por supuesto la clave en secreto. Este tipo de sistemas criptográficos basado en compartir una clave secreta se llaman sistemas *simétricos*. Una vez fijada la clave k se tiene una algoritmo de cifrado e_k y un algoritmo de descifrado d_k con la condición evidente de que para cualquier texto en claro m, se verifique que $d_k(e_k(m)) = m$. Obsérvese que para ponerse de acuerdo en la clave se debe usar un canal seguro de comunicación.

Se usan esencialmente dos tipos básicos de cifrado simétrico: el cifrado en *flujo* y el cifrado en *bloque*. Una vez generada la clave se transmite a los usuarios mediante un medio seguro, por ejemplo usando los criptosistemas de clave pública que se verán después. En el cifrado en flujo se cifra cada carácter combinándolo con una sucesión obtenida mediante un proceso *pseudo-aleatorio*.

En el cifrado en bloque se trabaja con bloques de texto en claro y texto cifrado de longitud dada n. En general se usan tamaños de clave $n = 64, 128$ o 256 bits. En el año 2000, el cifrado simétrico en bloques Rijndael fue seleccionado (en un concurso abierto sin precedentes) como el AES (*Advanced Encrytion Standard*) que substituyó, después de veinte años, al DES (*Data Encrytion Standard*) propuesto en su día por el NIST (*National Institute of Standards and Technology*) de EEUU. El Rijndael cifra el texto en claro en bloques de tamaño 128 bits y la longitud de su clave puede variar entre 128 y 256 bits. El Rijndael ha sido diseñado para resistir el criptoanálisis clásico (criptoanálisis lineal, diferencial, ...). Sin embargo el Rijndael posee una estructura muy algebraica y ataques usando estructuras algebraicos (bases de Gröbner, aproximaciones cuadráticas...), no han sido estudiados con todo detalle (aunque de momento los intentos de romperlo han fracasado).

Criptosistemas de clave pública

El principal problema de los criptosistemas de clave simétrica es la necesidad de tener un canal de comunicación seguro para intercambiar la clave usada. En 1976 Diffie-Hellman [3] introdujeron el concepto de *criptografía de clave pública*. Cada entidad tiene su clave $k = (pk, sk)$ que consiste en una clave privada sk y otra pública pk. La clave privada se debe mantener en secreto y la pública se debe hacer accesible a cualquier persona o entidad con la que se quiera establecer comunicación. La potencia de este sistema de claves reside en que cualquier cosa que se cifre con una de ellas se descifra con la otra. Si **A** quiere transmitir información m, de forma confidencial, a **B** lo que hace es usar la clave pública pk_{B} de B para cifrar la información $pk_{\mathrm{B}}(m)$ y tansmitir el texto cifrado a **B**. Para descifrar el mensaje, **B** usa su clave privada sk_{B}: $sk_{\mathrm{B}}(pk_{\mathrm{B}}(m)) = m$. No es necesario por tanto ningún contacto a priori entre **A** y **B**.

Si por otra parte **B** quiere estar seguro de que es **A** quien le transmite la información lo que se debe hacer es que **A** cifre el texto en claro con su clave privada $sk_{\mathrm{A}}(m)$ el receptor **B** debe obtener la clave pública de **A** (de cualquier servidor en el que esté publicada) y descifra el mensaje $pk_{\mathrm{A}}(sk_{\mathrm{A}}(m)) = m$. De esta forma se garantiza la autenticidad del emisor.

El sistema de clave pública más usado es el RSA [13] que esta basado en la dificultad computacional de factorizar un número n, suficientemente grande, en sus factores primos. En el RSA se elige $n = pq$ siendo p y q primos muy grandes. Un primer ataque al RSA consiste en factorizar el entero n. Los algoritmos de factorización de enteros y los tests de primalidad

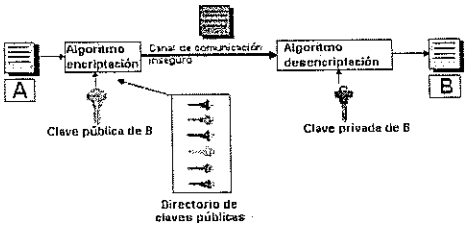

Figura 6: Esquema de confidencialidad.

(basados en teoría de números, en geometría algebraica,...) son, y han sido, aplicaciones de resultados matemáticos teóricos. Desde el punto de vista de la implementación, los algoritmos más eficientes de factorización son la criba cuadrática, las cribas de cuerpos de números (basado en cálculos en anillos de enteros algebraicos) y el algoritmo basado en curvas elípticas introducido por Lenstra. El último entero factorizado usado en RSA es de 576 bits. Este es el motivo de que en la actualidad se recomiende usar p y q de tamaño de 576 bits y n de 1024 bits como módulo, (incluso algunas agencias estatales proponen que n tenga 2048 bits). Desde el punto de vista teórico, cabe resaltar, el algoritmo AKS que permite saber si un entero es primo usando un algoritmo de complejidad polinomial. Sin embargo, por lo general, los ataques que más exitosos se deben al mal uso de los parámetros del RSA [8]. Para su implementación práctica matemáticos e ingenieros deben trabajar juntos en resolver el problema de encontrar la relación entre efectividad y seguridad.

Otro de los problemas computacionalmente difíciles que se usan en criptosistemas de clave pública es el *problema del logaritmo discreto*. En general, si $(G, \cdot)$ es un grupo cíclico finito de orden n generado por a, i.e. $G = \{a, a^2, \ldots, a^{n-1}, a^n = 1\}$, y dado $x \in G$ el problema consiste en encontrar k tal que $x = a^k$. Para que un grupo $(G, \cdot)$ se pueda usar en criptografía debe verificar las siguientes propiedades: i) los elementos del grupo deben ser expresados de un modo compacto, ii) la operación del grupo debe ser computada eficientemente, iii) el problema del logaritmo discreto debe ser intratable en G. En el año 1976, Diffie-Helmman establecieron un protocolo de intercambio de claves basado en el problema del logaritmo discreto sobre el grupo de unidades del cuerpo finito $\mathbb{F}_p$. En la actualidad y conociendo los resultados matemáticos actuales, los grupos que se usan en la práctica son los siguientes:

1) $(G, \cdot) = (\mathbb{F}_q^*, \cdot)$, donde $\mathbb{F}_q^*$ es el grupo de unidades del cuerpo finito $\mathbb{F}_q$ con $q = p$ con p primo o $q = 2^n$.

2) $(G, \cdot) = (E(\mathbb{F}_q), +)$, donde $E(\mathbb{F}_q)$ es el conjunto de puntos racionales de una curva elíptica sobre el cuerpo finito $\mathbb{F}_q$, con $q = p$ con p primo o $q = 2^n$.

El algoritmo más eficiente para resolver el problema del logaritmo discreto en los grupos de tipo 1) es el algoritmo del cálculo del índice, que es sub-exponencial. En general para los de tipo 2) no existen tales algoritmos y la complejidad de los algoritmos conocidos es exponencial. Esto hace que los sistemas basados en curvas elípticas proporcionen la misma seguridad con tamaños de clave mucho menores, por ejemplo una clave de curva elíptica de 163 bits proporciona la misma seguridad que otra de 1024 bits de RSA o de tipo 1). Tamaños de clave menores implican menor tiempo de proceso y menor espacio de almacenamiento, factores fundamentales para la aplicación práctica de la criptografía (por ejemplo en wireless, tarjetas electrónicas, ...), [7].

Otros grupos algebraicos (variedades abelianas sobre cuerpos finitos) donde el problema del logaritmo discreto es difícil y que se pueden usar en criptografía son los grupos de las jacobianas de curvas hiper-elípticas de género 2. De hecho se prueba que los criptosistemas son igual de eficientes que los de las curvas elípticas. El estudio, como herramientas de criptoanálisis, de formas bilineales no-degeneradas sobre curvas elípticas ha llevado a diseñar criptosistemas de clave pública basados en el problema de Diffie-Helmman usando formas bilineales no-degeneradas.

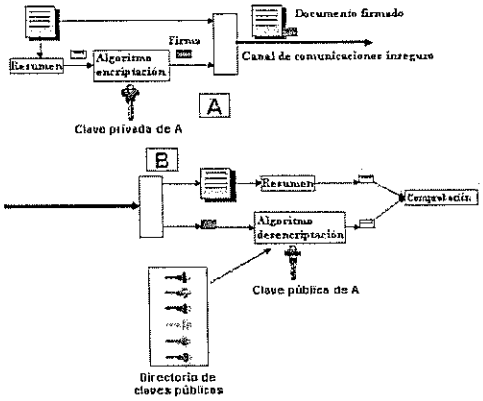

Figura 7: Esquema de Firma Digital.

A parte de la confidencialidad y la autenticidad, la criptografía debe garantizar la integridad y el no-repudio de los datos transmitidos. Tanto en los criptosistemas de clave simétrica como de clave pública hay métodos para garantizar la integridad (por ejemplo usando funciones *hash*). Los sistemas de clave pública, a través de los protocolos de *firma digital*, son capaces de garantizar la autenticidad, la integridad y el no-repudio. Este protocolo es de gran importancia en el mundo actual, tiene ámbito jurídico y legal y se usan para firmar contratos, realizar compras, hacer gestiones con la administración pública, entre otras muchas cosas. Como se puede ver en el esquema anterior, para realizar una firma de un texto en claro m, se procede de la siguiente manera: el firmante **A** hace un *hash* o resumen $h(m)$ de m (integridad), a continuación cifra el resultado con su clave privada (autenticidad, no-repudio) $sk_{\mathbf{A}}(h(m))$, lo añade al final del mensaje y transmite $(m, sk_{\mathbf{A}}(h(m)))$. Cuando **B** lo recibe, descrifra $pk_{\mathbf{A}}(sk_{\mathbf{A}}(h(m)))$, calcula $h(m)$ y comprueba que ambos coinciden, a esta fase se le denomina verificación de la firma. La seguridad de los algoritmos de firma digital es crucial para asegurar el futuro de las transacciones electrónicas.

Existen al menos dos problemas que comprometen seriamente tanto a los algoritmos de cifrado como al protocolo de firma digital y al nuevo DNI electrónico, mencionado al principio:

1.- Análisis criptográfico de las funciones hash. En la práctica los algoritmos de firma primero comprimen el texto en claro y luego cifran el texto comprimido. La finalidad de las funciones hash es limitar el tamaño de la firma independientemente de la longitud del texto en claro. Se pueden hacer protocolos probabilísticos de firma digital sin más que modificar la función hash usando sucesiones pseudoaletoria de bits. Una función *hash* $h : \{0,1\}^* \longrightarrow \{0,1\}^n$ es una función que comprime una cadena arbitraria de bits a una cadena de bits de longitud fija n. Las funciones hash no son inyectivas y se llaman *colisiones* a los $m, m' \in \{0,1\}^*$ tales que $h(m) = h(m')$. Para que una función hash se pueda usar en criptografía debe verificar: 1) fácil calcular $h(m)$ pero computacionalmente difícil calcular las preimágenes $h^{-1}(y)$, 2) debe ser computacionalmente difícil que fijado m se encuentre m' tal que $h(m) = h(m')$ y 3) debe

ser computacionalmente difícil encontrar colisiones. Las colisiones se pueden usar para alterar contenidos de documentos firmados. En el año 2005 se encontraron colisiones a la función hash SHA-1. Esta función fue desarrollada por la NSA de EEUU. y en la actualidad se usa como standard tanto a nivel oficial (gobiernos,..) como comercial (Microsoft, Sun, IBM..), [9].

El RSA fue el primer criptosistema de clave pública con protocolo de firma digital. El Gamal propuso un protocolo de firma digital no determinista (basado en el logaritmo discreto) que fue modificado ligeramente por el NIST dando lugar al DSA (Digital Stanandard Algorithm). Desde el año 2000 se está usando un algoritmo de firma digital ECDSA basado en curvas elípticas, que se está aplicando en el ámbito comercial y en las gestiones con las administraciones públicas. Todos estos protocolos usan el SHA-1 como función hash.

2.- Criptografía cuántica y post-cuántica. En el año 1994 Shor [14] encontró un algoritmo de factorización que en un "ordenador cuántico" tiene complejidad polinomial. Por tanto, si se llega a construir un ordenador cuántico de tamaño industrial todos los criptosistemas y protocolos basados en el problema de la factorización o en el problema logaritmo discreto serían inseguros. En la actualidad no se sabe si se llegará a construir un ordenador cuántico y cuando, pero dado el interés del tema hay muchos grupos de investigación que incluyen matemáticos trabajando en ello. Hay que decir a este respecto, que ya se usa criptografía basada en fenómenos cuánticos, incluso a nivel comercial, para la transmisión de información aunque no existan todavía ordenadores cuánticos.

Por todo lo anterior se deben proponer otros sistemas de clave pública no basados en estos problemas. Entre ellos cabe destacar: criptosistemas basados en problemas relacionados con retículos (por ejemplo NTRU, [6]), criptosistemas basados en teoría de códigos, criptosistemas basados en el problema de la conjugación en el grupo de trenzas [1], criptosistemas basados sistemas de ecuaciones polinomiales multivariables de grado dos sobre cuerpos finitos que usan algebra conmutativa y geometría algebraica, (por ejemplo HFE [12] ...). Para que cualquiera de estos protocolos puedan ser admitidos comercialmente es necesario un extenso trabajo de análisis y criptoanálisis en el que las matemáticas juegan un papel fundamental.

El autor quiere agradecer a Ignacio Luengo Velasco y a Julia Sánchez Meneses por sus discusiones muy interesantes y esclarecedoras sobre el tema.

Bibliografía

[1] I. Anshel, M. Anshel, B. Fisher, D. Goldfeld, *New key agreement protocols in braid group cryptography.* Topics in cryptology—CT-RSA 2001 (San Francisco, CA), 13–27, Lecture Notes in Comput. Sci., 2020, Springer, Berlin, 2001.

[2] R.F. Churchhouse, *Codes and ciphers : Julius Caesar, the enigma, and the Internet* Cambridge : Cambridge University Press, 2002

[3] W. Diffie, M. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory 22 (1976) 644–654.

[4] T. ElGamal, *A public key cryptosistem and a signature scheme based on discrete logarithms*, IEEE Transactions on Information Theory 31 (1985), 469–472.

[5] A.Fúster Sabater, D. de la Guía, L. Hernández, F. Montoya, J. Muñoz-Masqué, *Técnicas criptográficas de protección de datos*, Ed. Ra-ma, cop. 2004 Madrid.

[6] J. Hoffstein, J. Pipher, J.H. Silverman, *NSS: an NTRU lattice-based signature scheme.* Advances in cryptology—EUROCRYPT 2001 (Innsbruck), 211–228, Lecture Notes in Comput. Sci., 2045, Springer, Berlin, 2001.

[7] N. Koblitz, *Algebraic aspects of cryptography* Berlin, Springer, 1999 (segunda edición)

[8] N. Koblitz, A. Menezes, *A survey of public-key cryptosystems*, SIAM Rev. 46 (2004), no. 4, 599–634.

[9] S. Landau, *Find me a hash.* Notices Amer. Math. Soc. 53 (2006), no. 3, 330–332.

[10] A. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of applied cryptography*, With a foreword by Ronald L. Rivest. CRC Press Series on Discrete Mathematics and its Applications. CRC Press, Boca Raton, FL, 1997.

[11] R. Mollin, *An introduction to cryptography* Boca Raton : Chapman-Hall/CRC, cop. 2001.

[12] J.Patarin, *Asymmetric cryptography with a hidden monomial and a candidate algorithm for* $\simeq$ 64 *bits asymmetric signatures.* Advances in cryptology—CRYPTO '96 (Santa Barbara, CA), 45–60, Lecture Notes in Comput. Sci., 1109, Springer, Berlin, 1996.

[13] R.L. Rivest, A. Shamir, L. Adleman, *A method for obtaining digital signature and public key cryptosistem*, Communication of the ACM 21 (1978), 120–126.

[14] P.W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer* SIAM J. Comput. 26 (1997), no. 5, 1484–1509.

[15] D. Stinson, *Cryptography : theory and practice* Boca Raton, Chapman-Hall/CRC Press, cop. 2002

Alejandro Melle Hernández
Departamento de Álgebra,
Facultad de Matemáticas
Universidad Complutense de Madrid
CP 28040
Correo electrónico: amelle@mat.ucm.es