Los secretos de la matemática de los secretos

José L. Gómez Pardo
Departamento de Alxebra, Universidade de Santiago
15782 Santiago de Compostela, Spain
E-mail: pardo@usc.es

Prólogo

La criptografía es una ciencia muy antigua, cuyas primeras manifestaciones históricas se remontan al Antiguo Egipto y datan de hace más de 4000 años. Las matemáticas son todavía más antiguas y su origen se entremezcla con el de la humanidad como especie inteligente. Sin embargo, hasta hace muy poco tiempo ambas disciplinas no convergieron, un hecho que cambió profundamente la criptografía, que hoy en día puede, en gran medida, ser considerada como una disciplina matemática.

Una de las razones por las que la criptografía ha permanecido aislada de las matemáticas durante tanto tiempo es que, al ser su utilización tradicional la de comunicarse en secreto, fue hasta tiempo muy reciente un patrimonio casi exclusivo de las organizaciones de inteligencia que operan en los ámbitos militar y diplomático; de hecho, aun a principios de los años 70 del siglo pasado, la criptografía no existía como disciplina académica y era muy difícil adquirir conocimientos sobre la misma porque casi todo el material criptográfico que tenía algún interés estaba clasificado. Esta fue la situación con la que se encontraron los pioneros como W. Diffie, M. Hellman, R. Merkle y R. Rivest, que estaban interesados en la criptografía y en su uso civil pero encontraban todo tipo de dificultades e incluso amenazas cuando pretendían desarrollarla. No sólo la criptografía se usaba para guardar los "secretos de estado" sino que ella misma era un gran secreto. Hoy en día, las cosas han cambiado mucho y, aunque siguen existiendo organizaciones poderosas que mantienen sus conocimientos criptográficos en secreto, la criptografía es una disciplina con un gran desarrollo en el ámbito académico y, como he señalado antes, está estrechamente vinculada a las matemáticas. En estas notas, trataré de pasar revista a algunos de los aspectos más importantes de la criptografía moderna y, en particular, a la influencia de las matemáticas en la misma.

Un poco de historia

¿Qué es la criptografía? Etimológicamente, la criptografía es la *"escritura secreta"* y esto ya proporciona implícitamente una buena descripción de sus objetivos. Ampliando un poco esta definición se puede decir que la criptografía es la ciencia de la comunicación segura, siendo su objetivo más básico –aunque no el único, como más adelante veremos– el permitir que dos personas (a las que llamaremos Alicia y Bernardo, siguiendo la tradición iniciada en [19], donde estas personas son Alice y Bob) puedan comunicarse a través de un canal inseguro sin que un oponente (llamémosle Eva) que también tiene acceso a dicho canal, pueda comprender la información que está siendo transmitida. El método que se usa habitualmente para conseguir este fin es utilizar un *criptosistema*, el cual proporciona un algoritmo para que Alicia cifre la información que quiere transmitir (que recibe el nombre de

texto claro) utilizando una clave que previamente ha acordado con Bernardo. La información cifrada recibe el nombre de *criptotexto* y es enviada a través del canal. De esta forma Eva tiene acceso al criptotexto pero, como no conoce la clave, no puede recuperar el texto claro. Por el contrario, Bernardo, que sí la conoce, utiliza la clave para descifrar el criptotexto y recuperar el texto claro, mediante la operación inversa de la realizada por Alicia. El objetivo fundamental de la criptografía, considerada como una ciencia, es, por tanto, el diseño de criptosistemas seguros y eficientes.

A menudo, cuando se habla de criptografía, se utiliza el término con un significado mucho más amplio que el anteriormente expuesto. Paralelamente a la criptografía, existe también la disciplina que podríamos llamar "opuesta", la cual consiste en buscar los medios para *romper* o, en términos más técnicos *criptoanalizar* los criptosistemas, para así poder recuperar el texto claro o, incluso, la propia clave, sin partir del conocimiento previo de esta (esto es lo que, probablemente, trataría de hacer Eva en la situación antes descrita). Esta disciplina recibe el nombre de *criptoanálisis* y, a menudo, se da el nombre de criptografía a lo que en realidad debe llamarse *criptología*, que es la conjunción de la criptografía y el criptoanálisis. De hecho, la historia de la criptología puede ser vista como una lucha entre criptógrafos y criptoanalistas. La creencia popular casi siempre ha sido que los criptógrafos llevan las de ganar en esta lucha. Esta creencia fue bien explotada por un personaja famoso del siglo XVIII que narra en sus memorias como, después de descifrar un manuscrito que supuestamente contenía la fórmula para transmutar metales viles en oro y que una señora llamada Madame d'Urfé le había confiado, *"... me hice el dueño de su alma y abusé de mi poder ..."*. No nos consta que nuestro personaje haya aprovechado la ocasión para fabricar oro y es más que probable que no haya sido así pero queda claro que sí la aprovechó para seducir a la señora en cuestión, lo cual no resulta nada extraño si se tiene en cuenta que se trataba del aventurero veneciano Giacomo Casanova.

Por el contrario, la historia muestra que han sido siempre los criptoanalistas los que han terminado venciendo (aunque, como indicaré más adelante, hay indicios de que esta situación puede estar cambiando, precisamente debido al uso de las matemáticas en criptografía). La razón de ello ha sido bien expresada por Edgar Alan Poe en su relato "The Gold Bug" ("El escarabajo de oro",[18]), en el cual su protagonista, Will Legrand, descifra un criptograma de sustitución mediante el "análisis de frecuencias". Dice Legrand: *"es dudoso que el genio humano pueda crear un enigma de ese género que el mismo ingenio humano no resuelva con una aplicación adecuada"*. Pero, eventualmente, los criptógrafos llegarían a pensar: ¿y si no basáramos los criptosistemas o, al menos, no solamente, en el ingenio? ¿y si los basáramos en problemas matemáticos difíciles?

Pero las matemáticas hicieron su irrupción en el criptoanálisis antes que en la criptografía. El primer uso no trivial de las matemáticas en criptología se debe, muy probablemente, al criptoanalista norteamericano de origen ruso William Friedman, quien hacia los años 20 del siglo pasado, utilizó métodos estadísticos que aprovechaban la redundancia inherente a todos los lenguajes naturales (el *índice de coincidencia*) para criptoanalizar cifras de sustitución polialfabética como la de Vigenère y, en 1939, consiguió criptoanalizar la máquina japonesa *Purple*. En la primera mitad del siglo XX, la criptografía aun era usada casi exclusivamente por organizaciones gubernamentales para actividades relacionadas con la información y el espionaje, de modo que permanecía, en gran medida, aislada del mundo científico. Pero, poco a poco, estas organizaciones se fueron dando cuenta de que la criptología era susceptible de ser estudiada matemáticamente. Un momento muy importante, que marca un punto de inflexión en la historia del criptoanálisis, se produjo en los años 30, cuando Polonia se encontraba sometida a una fuerte presión ante las ansias ex-

pansionistas de la Alemania nazi. Los servicios de inteligencia polacos consideraban vital para los intereses de su pais el poder criptoanalizar la máquina *Enigma*, que los militares alemanes utilizaban para cifrar todas sus comunicaciones. En ese momento se acordaron de los matemáticos y en 1931 sometieron a una prueba a 20 de ellos, pertenecientes a la Universidad de Poznan –elegida por encontrarse en una zona de Polonia en la que se hablaba alemán–, a quienes previamente habían hecho jurar secreto. De los 20, tres mostraron grandes aptitudes criptoanalíticas y uno de ellos, que por entonces tenía sólo 23 años, iba a pasar a la historia de la criptología; no sólo su país le debe mucho, sino también mucha otra gente que ni siquiera sabe de su existencia. Se trataba de Marian Rejewski, quien en los años inmediatamente anteriores a la segunda guerra mundial consiguió criptoanalizar Enigma. Ello no evitó que Polonia sucumbiera ante el empuje del Tercer Reich, pero su trabajo fue continuado en Inglaterra, ya comenzada la guerra, por el llamado *grupo de Bletchley Park*, cuyo miembro más destacado era otro matemático, al que hoy en día se considera el padre de la moderna teoría de la computación: Alan M. Turing. La máquina Enigma había sido modificada y era ahora más segura pero, tomando como punto de partida el trabajo de Rejewski, este grupo consiguió criptoanalizar las nuevas versiones de la máquina, permitiendo la obtención de información que resultó decisiva para el ulterior desarrollo de la guerra.

La primera formalización sistemática de conceptos básicos de la criptografía en términos matemáticos se produce, ya terminada la segunda guerra mundial, con la publicación del trabajo [21] en 1949, en el que el fundador de la *teoría de la información* usó esta para estudiar los criptosistemas. En particular, Shannon consideró un modelo matemático para evaluar la seguridad de un criptosistema y consiguió demostrar la existencia de criptosistemas *incondicionalmente seguros* en el sentido intuitivo de ser resistentes al criptoanálisis, incluso ante un criptoanalista dotado de recursos computacionales ilimitados. Un tal criptosistema había sido introducido ya en 1917 por Gilbert Vernam, por lo que recibe el nombre de *cifra de Vernam*. Consiste, esencialmente, en utilizar una clave, elegida aleatoriamente, de la misma longitud que el mensaje. Por ejemplo, si el mensaje es una sucesión de bits, se genera aleatoriamente otra sucesión de la misma longitud y se combinan ambas mediante la operación XOR –suma de bits módulo 2– para producir el criptotexto. No sólo la clave tiene la misma longitud que el mensaje sino que para cada nuevo mensaje hay que utilizar una nueva clave, de ahí el nombre de *cuaderno de uso único* (one-time pad, en inglés) con el que también se conoce a este criptosistema. Resulta intuitivamente evidente que, dado un criptotexto producido con este sistema, cualquier texto claro que tenga la misma longitud tiene idéntica probabilidad de haberlo originado, dado el caracter aleatorio de la clave y, lo que hizo Shannon, fue demostrar rigurosamente este hecho. Más aun, Shannon demostró también que para tener seguridad incondicional un criptosistema debe de usar una clave cuya longitud no sea inferior a la del mensaje, de modo que la cifra de Vernam es, esencialmente, el único criptosistema con esta propiedad. Shannon definió la *"distancia de unicidad"* de un criptosistema como la menor cantidad de texto claro que puede ser descifrada de forma única a partir del criptotexto correspondiente, suponiendo que el atacante dispone de recursos ilimitados. Cualquier criptosistema es seguro si se utiliza sólo por debajo de su distancia de unicidad, pero este hecho tiene poco valor práctico si la distancia de unicidad es baja, y el único que tiene distancia de unicidad infinita es, precisamente, la cifra de Vernam. No obstante, la utilidad práctica de este criptosistema es muy limitada, debido a la necesidad de intercambiar a través de un canal seguro una clave de la misma longitud que el mensaje (si se dispone de un canal seguro para ello, ¿por qué no utilizarlo directamente para enviar el mensaje?). Aunque puede haber sido usada en el espionaje, es claro que esta cifra es completamente inadecuada en la mayoría de las situaciones como,

por ejemplo, en las relacionadas con el comercio electrónico.

A pesar del gran impulso que la teoría de Shannon dio a la "matematización" de la criptografía, la auténtica irrupción de las matemáticas en este campo estaba todavía por llegar. Para comprender las razones que llevaron a este cambio, conviene tener en cuenta que, además de la confidencialidad, hay otros objetivos muy importantes en la criptografía moderna, como son la *autenticidad* (cuando Bernardo recibe un mensaje de Alicia, sabe que es realmente Alicia quien lo envía); la *integridad* (Bernardo puede detectar si el mensaje que le ha enviado Alicia ha sido alterado por un tercero) y la propiedad de *no repudio* (después de haber enviado un mensaje a Bernardo, Alicia no puede afirmar que el mensaje no es suyo). La importancia de todos estos objetivos es fácil de comprender si se piensa, por ejemplo, en el intercambio de mensajes originado porque Alicia compre un artículo a Bernardo a través de Internet, realizando el pago con una tarjeta de crédito.

Los criptosistemas clásicos, en uso en los años 70, se basaban en el uso de una clave secreta que las dos partes compartían y que era la que se usaba tanto para cifrar como para descifrar (de ahí el nombre de *criptosistemas simétricos*, que también reciben). Estos criptosistemas estaban diseñados pensando en mantener la información secreta, pero carecían de mecanismos para alcanzar los restantes objetivos mencionados. Así, para que Bernardo pueda convencer a una tercera persona –por ejemplo, a un juez, en caso de disputa– de que un cierto mensaje que ha recibido procede de Alicia, se necesita una *firma digital*, que es el análogo electrónico de una firma escrita. Dado que en un criptosistema de clave secreta Alicia y Bernardo tienen la misma capacidad para cifrar y descifrar, Bernardo podría haber construido el mensaje y atribuírselo a Alicia. Resulta evidente que los criptosistemas simétricos no son adecuados, por sí solos, para proteger las comunicaciones entre múltiples personas que probablemente no se conocen y nunca antes han estado en contacto (como se producen habitualmente a través de Internet). El uso de una clave secreta requiere que esta haya sido compartida a través de un *canal seguro*, algo que no está fácilmente disponible en esa situación. A todo ello hay que añadir una dificultad adicional derivada del uso masivo que se hace hoy en día de la criptografía. En una red con n usuarios, un criptosistema clásico requiere que cada par de usuarios compartan una clave secreta, lo que implica un total de $n(n-1)/2$ claves y hace que, para n grande, su gestión se haga inmanejable.

Otro de los aspectos insatisfactorios de los criptosistemas clásicos era que, al no existir una sólida teoría matemática subyacente, no se disponía tampoco de criterios rigurosos para evaluar su seguridad, es decir, su resistencia al criptoanálisis, salvo en casos muy concretos como el de la cifra de Vernam. Además, después de los esfuerzos criptoanalíticos desarrollados en la segunda guerra mundial, se puede decir que todos los criptosistemas de importancia práctica que eran públicamente conocidos con anterioridad (obviamente, la cifra de Vernam no se incluye entre ellos), han sido rotos. Por ejemplo, no es difícil escribir un programa que, usando el índice de coincidencia, descifra automáticamente el criptosistema de Vigenère, que llegó a ser conocido como *"le chiffre indéchiffrable"* después de ser usado durante siglos sin que nadia pudiera romperlo; más adelante daré una idea de como se consigue esto. Otros criptosistemas simétricos más recientes también han sido rotos, siendo quizá el ejemplo más conocido y más importante el DES (Data Encryption Standard). El DES fue aprobado como estándar en 1975 por el National Bureau of Standards norteamericano, tomando como base el criptosistema *Lucifer* que había sido desarrollado por un equipo de IBM. Lucifer usaba una clave de 128 bits, pero la adoptada para el DES fue sólo de 56 bits (muy probablemente, por imposición de la NSA, la *National Security Agency*, que no quería que hubiese criptografía muy fuerte fuera de su control). Desde el principio, el DES fue muy controvertido, pues muchos criptógrafos pensaban que el tamaño de la clave

era insuficiente para prevenir un ataque criptoanalítico por fuerza bruta y, además, dado que los criterios seguidos para diseñarlo no se hicieron públicos, se sospechaba que la NSA podía haber forzado la introducción de una "puerta trasera" que le permitiría romperlo fácilmente. Pero nadie ha podido demostrar que tal puerta trasera existiese y, de hecho, se sabe ahora que el DES fue diseñado para resistir, por ejemplo, el "criptoanálisis diferencial" que no fue descubierto por la comunidad académica (Biham y Shamir) hasta principios de los años 90, pero que ya era conocido en los 70 por los criptólogos de IBM, que se vieron obligados a mantenerlo en secreto. Finalmente, el DES fue roto por fuerza bruta en 1997 y, en 1998, una máquina construida ex profeso, el "DES Cracker", desarrollado por la EFF (Electronic Frontier Foundation) y que contaba con 1536 chips, era capaz de romper el DES mediante búsqueda exhaustiva del espacio de claves en 4,5 días por término medio. De hecho, en enero de 1999, el DES Cracker, trabajando en colaboración con una red de unos 100.000 ordenadores de todo el mundo, fue capaz de encontrar una clave del DES en 22 horas y 15 minutos, comprobando a razón de más de 245 mil millones de claves por segundo. La obsolescencia del DES se produjo así como consecuencia de la severa limitación al tamaño de la clave impuesta por la NSA y no por fallos en el diseño del mismo, pues el mejor ataque criptoanaítico anti-DES conocido públicamente, el "criptoanálisis lineal" (Matsui, 1994), tiene muy poca importancia práctica al ser un ataque "con texto claro conocido" que requiere disponer de un gran número de pares formados por un texto claro y su correspondiente criptotexto (2^{43} en la implementación realizada por Matsui).

Lo anterior no significa que los criptosistemas simétricos sean inherentemente inseguros. El más conocido en la actualidad es el sucesor del DES: el AES (*Advanced Encryption Standard*]), que fue aprobado como estándar por el NIST (*National Institute for Standards and Technology*) en 2001, después de un concurso público al que fueron aceptados 15 candidatos provenientes de diferentes países. El AES (también conocido como *Rijndael*, un acrónimo de los nombres de sus autores) fue diseñado por los belgas Joan Daemen y Vincent Rijmen y es un *criptosistema de bloques* (como también lo es el DES), en el que el texto claro es dividido en bloques del mismo tamaño (de 128 bits en este caso) cada uno de los cuales se cifra como un bloque de criptotexto. En cuanto a la clave, puede ser de 128, 192, o 256 bits. El AES fue diseñado con el objetivo de que se comportase como una permutación seudoaleatoria y de forma que fuese resistente a todos los ataques criptoanalíticos conocidos en ese momento. A diferencia de otros criptosistemas anteriores (por ejemplo el DES), tiene una estructura algebraica subyacente bastante sencilla, que se basa en la aritmética del cuerpo de 256 elementos, $\mathbb{F}_{256}$ (lo que permitió a H.W. Lenstra describir el AES en una página, cf. [14]). Esto se debe en parte al deseo de disipar las sospechas de que la llamada "caja S", que en este caso se basa en la operación de inversión en $\mathbb{F}_{256}$, pueda contener una puerta trasera (las "cajas S" del DES parecían ser aleatorias, lo cual las hacía muy sospechosas) y también porque así se sabe que se dificultan algunos ataques conocidos (criptoanálisis lineal y diferencial). No expondré aquí los detalles del algoritmo del AES y remito a las referencias y, en particular, a mi implementación del AES en *Mathematica* [9], que contiene también implementaciones de métodos de autentificación basados en AES, a través del uso de MAC's (*Message Authentication Codes*) y, en particular, de OMAC, que es el modo que el NIST tiene intención de convertir en estándar próximamente.

En cuanto a la seguridad del AES, se puede decir que, por el momento, la gran mayoría de los expertos opinan que es seguro. Los ataques más importantes que se han ensayado sobre AES son conocidos como ataques algebraicos, pues tratan de explotar precisamente la estructura algebraica de Rijndael. Un intento notable en esta dirección es el de N. Courtois y J. Pieprzyk [5], quienes muestran que la caja S de Rijndael puede ser escrita como un sistema

sobredefinido de ecuaciones cuadráticas en varias variables de modo que, por ejemplo, la clave secreta de Rijndael de 128 bits puede ser recuperada a partir de un texto claro único mediante un sistema de 8000 ecuaciones cuadráticas con 1600 incógnitas binarias. Para que este planteamiento sea útil, es necesario disponer de un método que permita resolver este sistema para recuperar la clave. La propuesta de Courtois y Pieprzyk es usar una técnica conocida como *extended sparse linearization* (XSL) y es sobre este punto donde existe un cierto desacuerdo entre los expertos. La mayoría son totalmente escépticos y, entre ellos, algunos muy destacados como, por ejemplo, Don Coppersmith (quien se cree que fue el principal descubridor del criptoanálisis diferencial hecho por los criptólogos de IBM cuando el DES estaba siendo diseñado). Coppersmith cree que el método falla, pues los autores no disponen de un número suficiente de ecuaciones linealmente independientes para resolver el sistema. Los propios Courtois y Pieprzyk admiten que, aunque ellos creen que su ataque podría romper el AES de 256 bits (no el de 128 bits), no existe certeza sobre ello pues la complejidad de XSL es muy difcil de evaluar. De todas formas, incluso si Courtois y Pieprzyk están en lo cierto, esto no significa que el AES sea inseguro en la práctica. Cuando hablan de "romper el AES" están utilizando el término en su "sentido académico". Esto significa que el sistema se rompe si se encuentra lo que se llama un *"shortcut attack"*, es decir, un ataque criptoanalítico que requiere un trabajo inferior que un ataque por fuerza bruta. Naturalmente, la existencia de un "shortcut attack" no significa, ni mucho menos, que el sistema pueda ser roto en la práctica y esto explica la aparente paradoja de que Courtois y Pieprzyk creen que pueden romper el AES con clave de 256 bits y no el de 128 bits que, en la práctica, siempre será más vulnerable. Aunque pueda no tener una significación práctica inmediata, la no existencia de "shortcut attacks" conocidos es uno de los criterios de calidad más importantes para una cifra, ya que si un tal ataque existe, ello puede proporcionar un indicio de que la cifra puede llegar a ser rota antes de lo esperado en el futuro. Desde luego, en el caso del AES, incluso con una clave de 128 bits, un ataque por fuerza bruta es impensable por el momento.

Descifrando la cifra indescifrable

Históricamente, se constata que los criptosistemas clásicos han sido rotos precisamente en el momento en que mayor confianza existía en su seguridad por haber resistido los ataques durante un cierto tiempo. Así ocurrió con el criptosistema de Vigenère que, habiendo sido introducido por Blaise de Vigenère en el siglo XVI, no fue roto hasta mediados del siglo XIX, cuando fue criptonanalizado por el oficial prusiano Friedrich Kasiski. A continuación mostraré lo fácil que es de romper, usando los métodos introducidos por William Friedman, la cifra que en su día fue considerada indescifrable.

En primer lugar, conviene recordar que una "cifra de sustitución monoalfabética", consistente en aplicar a las letras del alfabeto una permutación dada, sustituyéndolas por sus imágenes mediante dicha permutación, se puede romper fácilmente usando el llamado "análisis de frecuencias". Este método se basa en que las frecuencias de los caracteres alfabéticos de los lenguajes naturales tienen una distribución característica que se puede conocer a base de analizar un número suficiente de textos. Por ejemplo, en inglés, las letras más frecuentes son, en orden decreciente, "e", "t", "a", con frecuencias relativas aproximadas 0,127, 0,091 y 0,082. En castellano, la situación es un poco diferente y las tres letras más frecuentes son "e", "a", "o", con frecuencias aproximadas 0,137, 0,125 y 0,087. En una cifra de sustitución monoalfabética, la correspondencia biunívoca existente entre letras del texto claro y las del criptotexto, hace que esta distribución de frecuencias sea la misma en ambos (aunque con

las letras intercambiadas). Así, analizando la distribución de frecuencias de los caracteres del criptotexto (y, en caso necesario, también de pares o ternas de caracteres consecutivos –digramas, trigramas–, etc.) y comparándolas con la distribución estándar correspondiente al lenguaje del texto claro, se puede romper fácilmente dichas cifras (suponiendo, claro está, que se dispone de una cantidad de criptotexto suficiente). Una buena introducción al criptoanálisis de estas cifras se puede obtener a través de ejemplos literarios, desde el ya mencionado "El escarabajo de oro", de E.A. Poe, a "La aventura de los bailarines", de Arthur Conan Doyle, donde Sherlock Holmes ejerce de criptoanalista con su habitual ingenio. Para dificultar el análisis de frecuencias, se inventaron las "cifras de sustitución polialfabéticas", en las cuales se utiliza una sucesión periódica de cifras de sustitución monoalfabéticas. Así, si dicha sucesión consta de, por ejemplo, cinco sustituciones monoalfabéticas (es decir, cinco permutaciones del alfabeto), la primera letra del texto claro, junto con las que ocupan los lugares 6, 11, ..., 5k+1, ... se cifran con la primera de ellas, las que ocupan los lugares 2, 7, ..., 5k+2, ... se cifran con la segunda, y así hasta las que ocupan lugares que son múltiplos de 5, que se cifran con la última de las sustituciones. En general, si se usan n sustituciones, una letra se cifra con la r-sima sustitución ($1 \leq r \leq n$) si el lugar que ocupa en el texto claro es congruente con r módulo n. La cifra de Vigenère es una sucesión periódica de n sustituciones, cada una de las cuales es una "cifra de César", es decir, una sustitución obtenida por desplazamiento cíclico, haciendo corresponder a cada letra la que está k posiciones más adelante en el alfabeto. En la cifra de Vigenère, esta sucesión de sustituciones viene dada habitualmente por una "palabra clave" (que es la clave secreta que compartirán Alicia y Bernardo). La longitud (es decir, el número de letras) de la palabra clave es el periodo de la cifra y cada una de estas letras corresponde a una cifra de César. Imaginemos, por ejemplo, que se usa como clave la palabra "CESAR". La letra C indica que la sustitución que hay que aplicar a los caracteres del texto claro que ocupan una posición congruente con 1 módulo 5 es el desplazamiento (cifra de César) que aplica la "A" en la "C", es decir, desplaza cada letra dos posiciones hacia adelante (cíclicamente, de modo que la "Y" va a la "A" y la "Z" a la "B"). A continuación, la "E" indica que a las letras del texto claro que ocupan en este una posición congruente con 2 módulo 5 se les aplica el desplazamiento que aplica la "A" en la "E", es decir, cada letra se desplaza cuatro posiciones hacia adelante en el alfabeto. Así se continúa hasta las letras que ocupan una posición múltiplo de 5 a las que se aplica el desplazamiento correspondiente a la "R".

En una sustitución polialfabética no se puede aplicar directamente el análisis de frecuencias como método de criptoanálisis, ya que el mismo carácter del texto claro se puede cifrar de forma distinta según la posición que ocupe y, recíprocamente, diferentes caracteres del texto claro pueden dar lugar al mismo caracter en el criptotexto. Sin embargo, la redundancia de los lenguajes naturales sigue haciendo posible el criptoanálisis. Friedman usó para ello el *índice de coincidencia* (IC) que, para un texto dado, se define como la probabilidad de que dos caracteres del texto tomados al azar resulten ser idénticos. El IC de un texto es tanto mayor cuanto menos uniforme sea la distribución de frecuencias de los caracteres. Esto es evidente si se piensa, por ejemplo, en un caso extremo como puede ser un alfabeto formado por sólo dos símbolos, digamos a, b. Resulta completamente obvio que el IC de un texto formado por 99 letras "a" y una "b" es mucho mayor que el de un texto que tenga el mismo número de aes que de bes. De hecho, si se tiene un texto de longitud n sobre un alfabeto de r letras cuyas frecuencias relativas son f_0, f_1, ..., f_{r-1}, se puede ver fácilmente que el IC del texto es:

$$\frac{\sum_{i=0}^{r-1} f_i(f_i - 1)}{n(n-1)}$$

Así si consideramos, por ejemplo, un alfabeto de 27 letras como el del castellano (sin incluir espacios ni signos de puntuación e ignorando las tildes), vemos que un texto sobre este alfabeto, en el que los caracteres sean generados aleatoriamente con distribución de probabilidad uniforme, tiene un IC esperado de $\frac{1}{27} \approx 0,037$. Sin embargo, si se tiene en cuenta la distribución de frecuencias típica del castellano y se aplica la fórmula anterior, se obtiene que un texto castellano tiene un IC esperado de 0,078 (mis cálculos dan un IC un poco inferior, con un valor un poco superior a 0,075. Naturalmente, este valor no se puede determinar con exactitud y depende de los textos utilizados. Lo que sí se puede calcular exactamente es el IC de un texto concreto, por ejemplo, el de la versión de Don Quijote del "Project Gutenberg", [10], que consta de 1.640.590 caracteres en el alfabeto de 27 letras, resulta ser 0,0746771).

En consecuencia, un criptotexto castellano correspondiente a una sustitución monoalfabética debe tener un IC próximo a 0,078 (suponiendo que se haya usado el alfabeto antes indicado), pero un criptotexto procedente de una sustitución polialfabética como es la cifra de Vigenère tendrá un IC más pequeño y, cuanto mayor sea el número de sustituciones empleado (es decir, el periodo de la cifra), más se aproximará al IC de un texto aleatorio que, como ya he señalado, sería de 0,037 para el alfabeto de 27 letras. Supongamos entonces que tenemos un criptotexto como el siguiente, que sabemos que ha sido obtenido a partir de un texto castellano mediante el sistema de Vigenère (en criptología siempre se supone que el criptoanalista conoce el algoritmo y otra información relevante como, por ejemplo, el lenguaje del texto claro, y lo único que ignora es la clave; esto se conoce como *Principio de Kerckhoffs*).

"DEQDUAPOUMDSXIOJQQÑQGDNGJSXIXQGWPFKWAZTJDÑPUFIJBVLDNQJKI
LWIWDIUUMZJZTWGDEQXIBVXFUOFTYADHAIUABJJALOUIÑZBJSIYSGEVZBJ
KMZJXRRHLGCUCOWSVENZUZQZESVQBDNCBOLTDRNJFMVUÑEVVNRCIVHT
OXRNTJKXBWYÑQQOGMTWYOXONIGWVWTODKQDCLJRXODGBJULJOÑVR
GNHOMMSCEPRBBUJTOGHDQNCULNZVIÑDFFGIAWWSSDEBUAYJXVWOFOVI
TQIQHHPUFTJBVLHUBXJZQÑIQWSNBJAUDLXDZQJNMUDMXUOÑQWCJBWSH
ZRQGWBDVEEOXBYZXRIQTJUZJBNRXODANDWOJRXNDBPFFIBDVMRHNJKS
DATWVJÑYJAXPLIVJRQGWBDVEEOXBJZXQBQDBGUSKXAXQCDNSUTQBTVS
DEUFIVIBKXDLSJUXQBHRQNCKWMSFSPIUUF"

La estrategia para criptoanalizar el texto sería la siguiente. En primer lugar calculamos el IC del criptotexto que, en este caso, resulta ser igual a 0,0413058. Esto ya nos dice que la cifra no puede ser una sustitución monoalfabética (que es un caso particular de la cifra de Vigenère, cuando el periodo es igual a 1). Pero esta información no basta, ni mucho menos, para criptoanalizar el texto. Para ello, necesitamos primero conocer la longitud de la clave. Si conseguimos averigüarla, el resto ya será fácil. En efecto, si la longitud de la clave es k, se divide el criptotexto en una colección de k "subcriptotextos", cada uno de los cuales ha sido obtenido utilizando la misma cifra de César. El primero de ellos estará formado por las letras que ocupan las posiciones 1, $k+1$, $2k+1$, ..., es decir, por las letras cuya posición es congruente con 1 módulo k. El segundo por las que ocupan una posición congruente con 2 módulo k y así sucesivamente, hasta llegar al k-simo, que consiste en las letras que ocupan una posición múltiplo de k (es decir, congruente con k módulo k). Para completar el criptoanálisis sólo hay entonces que aplicar análisis de frecuencias a cada uno de estos submensajes (que han sido obtenidos por sustituciones monoalfabéticas). El método sirve no sólo para una cifra de Vigenère sino también, más en general, para cualquier cifra de

sustitución polialfabética, aunque el análisis de frecuencias final es más fácil en el primer caso pues se puede combinar con una búsqueda exhaustiva debido al bajo número de claves posibles en una cifra de César.

Volviendo pues, al criptotexto anterior, el problema es como determinar la longitud de la clave. Para ello podemos ir dividiendo el mensaje cifrado en t submensajes en la forma anteriormente indicada, donde $t = 1, 2, \ldots$, hasta un valor que sea lo suficientemente grande como para pensar que la longitud de la clave es inferior a dicho valor. Para cada uno de estos valores de t, calculamos el IC de cada uno de los t submensajes y hallamos la media de todos ellos, obteniendo un único valor asociado a t (por ejemplo, para $t = 1$, el IC correspondiente es 0,0413058, como ya he indicado). Para $t = 2$, el IC que se obtiene como media del IC del submensaje formado por los caracteres que ocupan una posición impar y el IC del submensaje formado por los caracteres de posición par es 0,0484338. Para $t = 3$, se obtiene un IC de 0,042006. Lo que ocurrirá es que, si k es la longitud de la clave, el valor del IC obtenido para $t = k$ debe de estar próximo al IC del castellano, es decir a 0,078. La razón es que, en este caso, cada uno de los mensajes ha sido obtenido por una sustitución monoalfabética (una cifra de César) y tiene, por tanto, el mismo IC que el correspondiente submensaje del texto claro. Por el contrario, para los valores de t distintos de k, esto no ocurre, pues los caracteres de cada submensaje no han sido cifrados con la misma sustitución y, por tanto, los IC de estos submensajes se aproximarán más al de un texto aleatorio. Por tanto, cabe esperar que el valor de t que tenga asociado un IC mayor (y próximo a 0,078), sea precisamente la longitud k de la clave. Ya hemos visto que para $t = 1, 2$ el IC está muy por debajo del de un texto castellano. Los IC correspondientes a $t = 4, \ldots, 12$ son:

t	4	5	6	7	8	9	10	11	12
IC	0,0482	0,0519	0,0487	0,0417	0,0492	0,0401	0,0786	0,0412	0,0471

Esto indica, con bastante claridad, que la longitud de la clave debe ser $k = 10$. De hecho, si continuamos calculando IC's hasta $t = 20$, vemos que todos ellos se mantienen por debajo de $0,052$ excepto el que corresponde a $t = 20$, que vale $0,0784$. Como se puede obsevar, los IC's son algo más altos para los valores de t que no son relativamente primos con k y para los valores múltiplos de k están muy próximos al correspondiente al valor de k. Esto no plantea ningún problema grave porque, si pensásemos que la longitud de la clave es un múltiplo de la longitud real, lo que obtendríamos al final sería una clave obtenida yuxtaponiendo varias veces la clave real y el sistema de Vigenère así obtenido es equivalente. El único inconveniente es que esto podría dificultar el análisis de frecuencias final al hacer que el tamaño de los submensajes fuese menor.

Llegados a este punto, solo queda hacer el análisis de frecuencias de los 10 submensajes en que dividiríamos el criptotexto anterior, lo que se puede hacer fácilmente de forma automática mediante un programa adecuado. No daré los detalles para no alargarme, pero en este caso se obtiene que la clave es "DONQUIJOTE" y el texto claro correspondiente es:

"apenashabiaelrubicundoapolotendidoporlafazdelaanchayespaciosatierralasdoradashebrasde sushermososcabellosyapenaslospequeñosypintadospajarillosconsusarpadaslenguashabian saludadocondulceymelifluaarmonialavenidadelarosadaauroraquedejandolablandacamadel celosomaridoporlaspuertasybalconesdelmanchegohorizontealosmortalessemostrabacuando elfamosocaballerodonquijotedelamanchadejandolasociosasplumassubiosobresufamosocaballo rocinanteycomenzoacaminarporelantiguoyconocidocampodemontiel"

Criptografía de clave pública y RSA

Hasta ahora he mencionado solamente criptosistemas simétricos, pero en la actualidad existen otros completamente distintos, que resuelven muchos de los inconvenientes anteriormente citados. Con la idea de superar dichos inconvenientes, W. Diffie y M. Hellman publicaron en 1976 su artículo *"New directions in cryptography"* [6], que iba a provocar un cambio revolucionario en el desarrollo de la criptografía, al mostrar que su uso era posible sin un intercambio previo de claves secretas. Además, este artículo situó la criptografía en el ámbito académico, liberándola del control casi absoluto que sobre ella habían ejercido hasta entonces las organizaciones secretas y enmarcándola de lleno en el ámbito de las matemáticas. Una de las ideas clave que subyacen en la propuesta de Diffie y Hellman es la de construir criptosistemas cuyo criptoanálisis sea, en la medida de lo posible, equivalente a la resolución de un problema matemático difícil. Aunque existen muchos problemas que no sabemos resolver, la mayoría de ellos no son adecuados para este propósito y la idea fue utilizar *problemas computacionales* difíciles, en el sentido de que, aun conociendo algoritmos para resolverlos, no podemos hacerlo por no ser factible ejecutarlos en tiempo razonable. Esto conlleva cambiar el modelo de seguridad incondicional utilizado por Shannon, por otro basado en la *seguridad computacional*: el concepto de no factible manejado por Shannon, que significaba *matemáticamente imposible, con independencia de los medios disponibles* se debe sustituir por el de *computacionalmente no factible* cuyo significado es totalmente distinto. En palabras del criptográfo Gilles Brassard, el usuario de un criptosistema no debe esperar ya que el criptoanalista no tenga *información* suficiente para romperlo, sino que no tenga *tiempo*. El uso del modelo computacional también permite un análisis más riguroso de la seguridad de un criptosistema en términos de la complejidad de los algoritmos conocidos para criptoanalizarlo (evaluada mediante la teoría de la *complejidad computacional*, que proporciona estimaciones de como crece asintóticamente el tiempo de ejecución de los algoritmos al ir creciendo el "tamaño del input"). Este análisis no proporciona al criptógrafo tranquilidad absoluta, pues siempre queda abierta la posibilidad de que se puedan descubrir nuevos algoritmos más eficientes, pero sitúa el problema de la seguridad de los criptosistemas en el ámbito matemático, con la mayor fiabilidad que ello conlleva. Por ejemplo, esto permite introducir el concepto de *seguridad demostrable*, la cual se obtiene cuando se puede demostrar que criptoanalizar un determinado sistema se reduce a un problema bien conocido y que se cree difícil. Por supuesto que una tal demostración de seguridad es relativa a otro problema y no absoluta, pero puede proporcionar un nivel de confianza mucho mayor que las meras apariencias de dificultad que exhibían muchos de los criptosistemas clásicos.

La propuesta concreta de Diffie y Hellman fue basar los criptosistemas en el concepto de *función de dirección única*, una función $f : X \to Y$ tal que $f(x)$ es (computacionalmente) "fácil" de calcular para cada $x \in X$ pero, para la mayoría de los $y \in Y$, es "difícil" calcular $f^{-1}(y)$. La idea es que "difícil", en este contexto, significa "computacionalmente no factible", es decir, no factible usando los mejores algoritmos conocidos y el mejor hardware. El uso de funciones de dirección única como funciones de cifrado no es suficiente para poder construir criptosistemas que cumplan el requisito de confidencialidad, pues si se usa una función de dirección única para cifrar, entonces ni siquiera el destinatario legítimo del mensaje sería capaz de descifrarlo. Por eso es interesante el concepto de *función de dirección única con trampa*, que es una función de dirección única para la cual existe una información adicional que permite invertirla de modo eficiente. Basándose en estos conceptos, Diffie y Hellman introdujeron los llamados *criptosistemas de clave pública* o *criptosistemas asimétricos*, que consisten en una familia $\{f_k : \mathcal{M} \to \mathcal{C} | k \in \mathcal{K}\}$ de funciones de dirección única con trampa.

Para cada $k \in \mathcal{K}$ debe de ser posible describir un algoritmo para calcular f_k tal que no sea factible obtener a partir de él un algoritmo para invertir f_k, a menos que se conozca la trampa correspondiente a k. Para usar el criptosistema, Alicia elige un $a \in \mathcal{K}$ aleatorio y publica en un directorio el "algoritmo de cifrado" E_A que calcula f_a, el cual constituye su *clave pública*. A partir de a también obtiene la trampa que permite invertir f_a mediante el "algoritmo de descifrado" D_A, pero no la hace pública pues esta es su *clave privada*. Si Bernardo quiere enviar a Alicia un mensaje $m \in \mathcal{M}$, busca en el directorio público la clave pública E_A de Alicia y calcula $c = f_a(m) \in \mathcal{C}$, que es el criptotexto que le envía. Dado que Alicia es la única persona que conoce la trampa que permite invertir f_a, solamente ella puede recuperar el mensaje m, mediante el algoritmo D_A. Es importante observar que los usuarios ya no necesitan intercambiar claves antes de comunicarse, con lo que los problemas antes mencionados en relación con la distribución y el manejo de las claves quedan automáticamente resueltos. Pero las ventajas de un criptosistema de clave pública van más allá, pues ahora sí son posibles las firmas digitales. Para ello suponemos que $\mathcal{M} = \mathcal{C}$ y si Alicia quiere enviar a Bernardo el mensaje firmado m, lo que hace es enviarle, junto con m, la "firma" $s = f_a^{-1}(m)$. Entonces Bernardo usa la clave pública de Alicia para calcular $f_a(s) = f_a(f_a^{-1}(m)) = m$. Es claro que solamente Alicia puede haber calculado s a partir de m, pues sólo Alicia tiene capacidad para invertir f_a y así Bernardo se convence de que el mensaje procede efectivamente de Alicia. En este esquema no hay secreto, pues Alicia ha enviado a Bernardo el mensaje m sin cifrarlo. Si se quiere obtener simultáneamente confidencialidad y firma digital, lo que hace Alicia es enviar a Bernardo $f_b(m)$ y $f_b(s)$.

El primer criptosistema de clave pública, y quizá el que más éxito ha tenido hasta la fecha, fue propuesto en 1978, dos años después de la publicación del artículo de Diffie y Hellman, por Rivest, Shamir y Adleman [19], razón por la cual es conocido con el nombre de RSA, que son las iniciales de sus inventores.

El criptosistema RSA se basa en la hipótesis, no demostrada pero altamente plausible, de que, para n y e enteros positivos dados, donde n es el producto de dos primos grandes, la función $m \mapsto m^e \pmod{n}$ es de dirección única con trampa. La trampa que permite invertir fácilmente la función es, precisamente, el conocimiento de los factores primos de n, por lo que se puede decir que RSA está basado en la dificultad del problema de la factorización de números enteros. Cada usuario U de RSA construye su clave (o su software lo hace por él) de la manera siguiente. En primer lugar U construye un entero $n_U = p_U q_U$ multiplicando dos primos grandes p_U, q_U de, aproximadamente, el mismo tamaño pero que no estén demasiado próximos (para dificultar al máximo la factorización de n_U). A continuación, calcula $\phi(n_U) = (p_U - 1)(q_U - 1)$ y elige un entero e_U tal que $1 < e_U < \phi(n_U)$ y $\mathrm{mcd}(e_U, \phi(n_U)) = 1$. Finalmente, U calcula el inverso multiplicativo de e_U módulo $\phi(n_U)$, es decir, el único entero d_U tal que $1 < d_U < \phi(n_U)$ y $e_U d_U \equiv 1 \pmod{\phi(n_U)}$. Todos estos cálculos se pueden hacer con facilidad utilizando algoritmos conocidos y, en particular, el cálculo de d_U se puede realizar mediante el algoritmo de Euclides. Una vez completado este proceso, U hace público el par (n_U, e_U), que constituye su *clave pública*, pero se guarda para sí d_U, que es su *clave privada*. n_U recibe el nombre de *módulo*, e_U es el *exponente de cifrado* y d_U el *exponente de descifrado*.

Para enviar un mensaje a U sólo hay que conocer su clave pública, si el texto claro es m (donde m es un entero menor que n_U), este se cifra calculando:

$$c = m^{e_U} \pmod{n_U}$$

U, por su parte, descifra el mensaje haciendo uso de su clave privada y calculando:

$$c^{d_U} \pmod{n_U} = m$$

La razón de que las funciones de cifrado y descifrado sean inversas, en este caso, es un teorema de Euler según el cual, para todo entero m tal que $\text{mcd}(m, n) = 1$, se verifica que $m^{\phi(n)} \equiv 1 \pmod{n}$.

Como ya he indicado, se cree que la función de cifrado que proporciona el criptotexto c a partir de m es una función de dirección única con trampa. Si se conoce la factorización de n_U en producto de primos, es fácil invertir esta función y descifrar, pues entonces se pueden obtener $\phi(n_U)$ y luego d_U de la misma forma que lo ha hecho U. Aunque no está demostrado, se cree que criptoanalizar RSA tiene esencialmente el mismo grado de dificultad que factorizar el módulo. Por tanto, es muy importante tener una idea de la dificultad del problema de la factorización para poder saber hasta que punto RSA es un criptosistema seguro. Para ello hay que analizar la complejidad de los algoritmos de factorización conocidos y, los mejores de ellos, son *algoritmos de tiempo subexponencial*. Estos algoritmos son más lentos que los de *tiempo polinómico* (cuando el tiempo de ejecución crece, a lo sumo, proporcionalmente a una función polinómica del tamaño del input que, en este caso, es el número de dígitos del entero a factorizar) pero más rápidos que los de *tiempo exponencial* (para los que este tiempo crece exponencialmente). En concreto, el más potente de los algoritmos de factorización conocidos, que recibe el nombre de *criba del cuerpo de números* ("Number Field Sieve", NFS) tiene complejidad subexponencial y, tomando como referencia la factorización de un módulo de RSA de 512 bits que requirió, en 1999, 5,2 meses de trabajo a varios cientos de ordenadores, se puede estimar que factorizar un módulo de RSA de 1024 bits con los mismos recursos requiriría más de 3 millones de años [7]. La factorización record en este momento es la de RSA-576, un número de 576 bits y 174 dígitos decimales perteneciente al concurso *The New RSA Factoring Challenge* [20]. Este concurso, con premios en metálico, fue establecido por la compañia RSA Security (heredera de la empresa original, RSA Data Security, fundada por los inventores de RSA) para monitorizar los avances en factorización que se van produciendo. Sigue una tradición instaurada ya en 1977, cuando se dio a conocer RSA al público en general, en un artículo de Martin Gardner publicado en la revista *Scientific American*. En dicho artículo aparecía como un desafío, con un premio de 100 dólares a quien lo lograra, la factorización de RSA-129, un número de 129 dígitos, que proporcionaba la clave privada para descifrar un mensaje cifrado mediante RSA. El mensaje y la clave habían sido construidos por los inventores de RSA, que eran también los que ofrecían el premio. Aunque en aquel momento se pensó que la factorización de este número tardaría billones de años, debido fundamentalmente a un error de cálculo de Ron Rivest, que más tarde él mismo admitió, la factorización fue completada en 1994 por D. Atkins, M. Graff, A.K. Lenstra y P.C. Leyland, con la colaboración de más de 600 investigadores procedentes de más de 20 países, siendo el método de factorización usado la *Criba cuadrática multipolinómica* (MPQS). Como resultado obtuvieron el texto claro siguiente:

"The magic words are squeamish ossifrage".

Esta fue la primera de las grandes factorizaciones de módulos de RSA realizadas mediante la colaboración de muchas personas y ordenadores de distintos países, facilitada por el desarrollo de Internet. Como he mencionado ya, el último número de este tipo que se ha conseguido factorizar es RSA-576 y el resultado es:

RSA-576 =

188198812920607963838697239461650439807163563379417382700763356422988859715234665485319060606506504743045317388011303396716199692321205734031879550656996221305168759307650257059

=

398075086424064937397125500550386491199064362342526708406385189575946388957261768583317

×

472772146107435302536223071973048224632914695302097116459852171130520711256363590397527

Esta factorización fue anunciada por Jens Franke el 2 de diciembre de 2003 y fue realizada por un grupo de investigadores alemanes y de varios otros países, usando NFS y ordenadores de diversas universidades y otros centros de investigación. El premio que obtuvieron era de 10.000 dólares y los números que aun quedan por factorizar van desde RSA-640 (de 640 bits) a RSA-2048 (de 2048 bits), con premios que oscilan entre 20.000 y 200.000 dólares. Comparando la tabla de los tamaños de los números a factorizar con la de los premios ofrecidos, se puede observar que estos últimos crecen de forma esencialmente lineal, muy por debajo del crecimiento subexponencial del tiempo requerido por NFS. Puede parecer así que los de RSA Security han sido un poco "rácanos" y que cada vez estas factorizaciones serán menos rentables económicamente para los que las realicen pero, también hay que tener en cuenta la *ley de Moore*, que durante los últimos 30 años ha proporcionado un crecimiento exponencial de la potencia computacional de las máquinas (postula que la densidad de transistores se duplica cada 18 o 24 meses, lo que esencialmente lleva a que se doble la potencia computacional en este tiempo) y, sobre todo, la posibilidad de que se llegue a descubrir un algoritmo de factorización realmente eficiente (por ejemplo, de tiempo polinómico). De hecho, tal algoritmo ya existe (como demostró Peter Shor [22]), pero necesita, para ser ejecutado, un ordenador cuántico y estas bestias existen sólo a nivel experimental y no está claro si llegarán a ser algún día una realidad (¡por una vez, el software va por delante del hardware!). De todas formas, actualmente se cree que un módulo de RSA de 2048 bits (unos 620 dígitos decimales), ofrece un nivel de seguridad muy alto y probablemente lo seguirá ofreciendo durante bastantes años y, si llegara pronto a ser factorizado, de lo que menos tendría que preocuparse RSA Security es de los 200.000 dólares del premio. Es conveniente también indicar que, en el mundo real, pueden existir ataques que, sin comprometer el algoritmo en que se basa un criptosistema (por ejemplo, en el caso de RSA, sin factorizar el módulo) pueden no obstante explotar debilidades del protocolo o de la implementación para romperlo. Un ataque de este tipo contra RSA es el *timing attack* descubierto por Paul Kocher en 1995, cuando era un estudiante en Stanford. Kocher demostró que es posible descubrir el exponente de cifrado midiendo cuidadosamente los tiempos de computación de una serie de descifrados. La idea subyacente es que, en el algoritmo de exponenciación binaria, que se usa tanto para cifrar como para descifrar en RSA, el tiempo requerido es mayor cuando en el exponente hay un bit "1" que cuando hay un bit "0". Se trata pues de un "ataque físico" que, aunque puede ser prevenido fácilmente por diversos métodos (por ejemplo, añadir "retardo" para que la exponenciación binaria requiera siempre un tiempo fijo) mostró en su día la existencia de una debilidad inesperada en muchas implementaciones del sistema. Otro ataque que tuvo éxito y que, más que contra RSA es contra un protocolo de implementación del mismo, es el ataque de Daniel Bleichenbacher contra PKCS #1, que es un ataque con "criptotexto elegido" contra este protocolo, utilizado en SSL ("Secure Sockets Layer"), el cual, a su vez, es usado por los navegadores web para poder realizar intercambios de información seguros. Tampoco expondré aquí los detalles de este ataque, pero sí mencionaré que se basa en enviar a un servidor alrededor de un

millón de mensajes construidos cuidadosamente y observar las variaciones en las respuestas (mensajes de error) del servidor. Esta debilidad ya ha sido corregida en las versiones más recientes de PKCS, usando OAEP, un protocolo que no permite crear un criptotexto válido sin conocer ya el texto claro correspondiente.

Un poco más de historia

La invención de los criptosistemas de clave pública condujo a una "matematización" profunda de la criptografía, en la que la teoría de números juega un papel preponderante porque proporciona muchas posibles funciones de dirección única. Puede ser interesante abordar la cuestión de por qué hubo que esperar tanto tiempo para que surgiese este tipo de criptografía, lo cual resulta sorprendente si se tiene en cuenta que las matemáticas que se usan en la implementación de RSA –aunque no así las que se usan para atacarlo, que son mucho más sofisticadas– provienen del siglo XVIII y ya eran bien conocidas por Euler (a excepción de algunos de los *tests de primalidad* usados para obtener primos de tamaño suficiente). Cabe pues preguntarse –como hizo Neal Koblitz en [13]– por qué Euler no inventó RSA para regalárselo a su protectora, Catalina II de Rusia (más conocida como *Catalina la Grande*) quien, siendo una mujer extremadamente inteligente, podría haber apreciado su valor. Una razón muy importante para que no haya ocurrido así salta a la vista. La criptografía de clave pública no tiene demasiado interés en el ámbito militar y diplomático, que se rige por estructuras jerárquicas y utiliza *agentes* que permiten un uso eficiente de la criptografía clásica de clave secreta. De hecho, incluso antes de la publicación de los artículos de Diffie-Hellman y Rivest-Shamir-Adleman, tanto la criptografía de clave pública como una versión un poco diferente de RSA fueron descubiertos en Inglaterra por científicos que trabajaban para los servicios secretos británicos [23]. Pero estos mantuvieron el descubrimiento en secreto hasta hace pocos años por lo que este hecho ha quedado relegado a una mera curiosidad, sin influencia alguna en la historia de la criptografía. Lo peor del caso, es que parece haber evidencia que muestra que los servicios de inteligencia británicos no llegaron a darse cuenta de la importancia del descubrimiento hasta después de que la criptografía de clave pública saliera a la luz en el campo académico, aunque esto puede también explicarse por el hecho que acabo de mencionar: este tipo de criptografía no era muy interesante para sus fines. Una hipótesis muy plausible es que la criptografía de clave pública nació hacia el último cuarto del siglo XX como respuesta a las necesidades planteadas por la creciente informatización de la economía. En la actualidad, grandes cantidades de dinero son transferidas electrónicamente a diario entre entidades financieras y, si dichas transferencias no estuviesen protegidas criptográficamente, sería posible que terceras personas pudieran obtener información sobre las mismas e, incluso, robar los fondos electrónicamente. También es evidente la necesidad de proteger diversos tipos de información confidencial que permanece almacenada en ordenadores a los que se puede acceder a través de redes informáticas: informes de empresas, datos personales, etc. Antes de los años 70 los informes de este tipo tenían forma impresa y se guardaban en lugares físicamente protegidos, pero ahora no es así y, de hecho, aunque muchas veces no sea de forma consciente, cuando navegamos por Internet estamos usando criptografía de clave pública todos los días. Otra razón importante para explicar el tardío descubrimiento de la criptografía de clave pública es el hecho de que, antes de la llegada de los ordenadores, todos los cálculos se hacían a mano (así es como trabajaba, por ejemplo, Euler, que tenía una gran habilidad con los cálculos y los algoritmos). En este contexto, es difícil imaginar el uso manual de RSA, que sería terriblemente engorroso aun partiendo de la base de que se trabajaría con números mucho más pequeños que los que las máquinas

utilizan actualmente. Las razones expuestas tienen un elemento común subyacente, que es la irrupción de la informática en nuestra vida cotidiana; antes de que este fenómeno se produjera se puede decir que no había necesidad de criptografía de clave pública.

Por último, cabe mencionar también otra razón, que está relacionada con la que acabo de exponer. Se trata de que, hasta tiempos relativamente recientes, los aspectos algorítmicos y computacionales de las matemáticas permanecían relegados a un segundo plano, quizá como consecuencia también de la dificultad de realizar cálculos útiles manualmente. Un ejemplo significativo lo proporciona el *teorema fundamental de la aritmética*, ya conocido por Euclides, que asegura que todo número entero se factoriza, en forma esencialmente única, como un producto de números primos. Durante muchos siglos esta cuestión permaneció cerrada, pues se conocía la existencia de la factorización y se conocían también algoritmos para obtenerla como, por ejemplo, ir probando a dividir por todos los primos menores que la raiz cuadrada del entero a factorizar. Dado que los enteros que se podían factorizar a mano iban a ser, en cualquier caso, muy pequeños, a nadie le importaba que este método fuese muy poco eficiente y no se percibía la necesidad de encontrar otros algoritmos que fuesen más rápidos. Es cierto que Gauss manifestó en las *Disquisitiones Arithmeticae* que el problema de la factorización era muy importante, pero no parece que haya dedicado esfuerzo real a estas cuestiones y, desde luego, no podía ni siquiera imaginar el carácter profético que tenían sus palabras. Algunos matemáticos, entre los que cabe destacar a Fermat, tuvieron ideas importantes sobre el problema. Fermat estableció el punto de partida básico de muchos métodos de factorización modernos, incluyendo NFS, pero esta fue para él una cuestión marginal, y el problema nunca fue realmente objeto de atención preferente por matemáticos importantes hasta que la llegada de los ordenadores, y la posibilidad que brindaban de realizar cálculos muy "grandes", hizo entrever el interés que tendría obtener algoritmos más eficientes, aun antes de que existiese realmente una aplicación práctica del problema. Esta aplicación llegó, finalmente, con el criptosistema RSA, que ha puesto el problema de la factorización de enteros en primera línea, hasta el punto de que en palabras de Crandall y Pomerance [4], el teorema fundamental de la aritmética ("la factorización existe") ha dado lugar en la actualidad al *problema fundamental de la aritmética* ("hallarla").

Criptografía basada en curvas elípticas

Después de la revolución propiciada por Diffie y Hellman, el avance de la criptografía no se ha detenido –más bien al contrario– y, como vamos a ver, existen criptosistemas que utilizan herramientas matemáticas mucho más avanzadas que las de RSA. Uno de los problemas computacionales en el que se basan algunos de estos criptosistemas es el llamado *problema del logaritmo discreto*. Este problema es la base del *Intercambio de claves de Diffie-Hellman*, que fue propuesto ya en [6] y que, si bien no es un criptosistema en el sentido estricto del término, pues no permite enviar información confidencialmente, tiene el objetivo de permitir a Alicia y Bernardo intercambiar una clave a través de un canal público sin que Eva, que observa la transmisión, pueda acceder a dicha clave. De esta forma, Alicia y Bernardo pueden después usar esa clave para intercambiar información mediante un criptosistema simétrico de clave secreta. La idea de la que partieron Diffie y Hellman es que, dado un primo p y un entero g convenientemente elegidos, la función $x \mapsto g^x \pmod{p}$, donde x es un entero positivo, es de dirección única. Para precisar un poco más, recordemos que si G es un grupo y $g \in G$, el *orden* de g es el menor entero positivo n tal que $g^n = 1$ (donde 1 denota el elemento neutro, o identidad, de G; si no existe un tal n, se dice que el orden de g es infinito). En lo sucesivo supondremos que g

tiene orden finito n, en cuyo caso $< g >= \{1, g, g^2, \ldots, g^{n-1}\}$ es un grupo de n elementos (se dice entonces que $< g >$ es el subgrupo cíclico de G generado por g; también se dice que $< g >$ tiene orden n pues se llama orden de un grupo finito a su número de elementos). Se puede identificar $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ con el grupo aditivo de las clases residuales (mód n) y la aplicación $\mathbb{Z}_n \to < g >$ dada por $x \mapsto g^x$ es una biyección que satisface $g^{x+y} = g^x g^y$ o, en otras palabras, un isomorfismo entre el grupo aditivo $\mathbb{Z}_n$ y el grupo multiplicativo $< g >$. El isomorfismo inverso se suele denotar por $\log_g$ y si $h \in < g >$, $\log_g h$ recibe el nombre de *logaritmo discreto* (o índice) de h en la base g; obviamente se tiene que $\log_g(h_1 h_2) = \log_g h_1 + \log_g h_2$ para $h_1, h_2 \in < g >$. El *problema del logaritmo discreto* (PLD) en un grupo G es el problema de, dados $g, h \in G$, hallar $\log_g h$ si existe (es decir, si $h \in < g >$) y lo que Diffie y Hellman pensaron es que el cálculo del logaritmo discreto en el grupo multiplicativo $\mathbb{Z}_p^*$ de las clases residuales no nulas módulo un primo, no es factible si p está bien elegido. Por el contrario, el cálculo de g^x, dados $g \in G$ y x, es muy fácil empleando el algoritmo conocido como exponenciación modular o exponenciación binaria (cf. [4]) y suponiendo que, como es bastante habitual, existe un algoritmo eficiente para efectuar la multiplicación en G algo que, desde luego, ocurre en caso de ser $G = \mathbb{Z}_p^*$. En otras palabras, la función $x \mapsto g^x$ es, en este caso, de dirección única. Obsérvese que, si G es finito de orden primo, el orden n de $g \in G$ divide al orden de G por el *teorema de Lagrange* y así, G es cíclico y cualquier elemento $g \neq 1$ de G es un generador y, por tanto, todo elemento de G tiene un logaritmo discreto en la base g.

Además del Intercambio de Diffie-Hellman, existen criptosistemas (capaces de proporcionar confidencialidad y/o firmas digitales) basados en el PLD, siendo quizás los más conocidos el llamado *criptosistema de El Gamal* y, en lo que a firmas digitales se refiere, el DSA (Digital Signature Algorithm). No describiré aquí estos sistemas, que están explicados con detalle en [8, 24, 25].

En tiempos recientes, ha habido un importante descubrimiento criptográfico originado por la idea de aplicar sistemas como los anteriores (Diffie-Hellman, El Gamal, DSA) en grupos para los que sólo se conocen algoritmos de complejidad exponencial para resolver el PLD. El salto de algoritmos subexponenciales a algoritmos exponenciales es muy grande y hace que estos nuevos sistemas gocen, en principio, de un nivel de seguridad mucho más elevado o, desde un punto de vista ligeramente diferente, que se pueda conseguir un nivel de seguridad dado con recursos computacionales mucho más limitados (usando claves mucho más pequeñas). La idea de estos nuevos criptosistemas se le ocurrió, independientemente, a Victor Miller y Neal Koblitz, en el año 1985 y, básicamente, consiste en usar el PLD sobre los grupos de las curvas elípticas.

Las curvas elípticas son, como su nombre indica, objetos geométricos, pero tienen una estructura algebraica (son grupos abelianos con respecto a la suma de puntos) que permite usarlas para definir, por ejemplo, el *criptosistema de El Gamal elíptico*. Este sistema está basado en el PLDE (problema del logaritmo discreto elíptico) es decir, en el PLD sobre el grupo de los puntos de una curva elíptica.

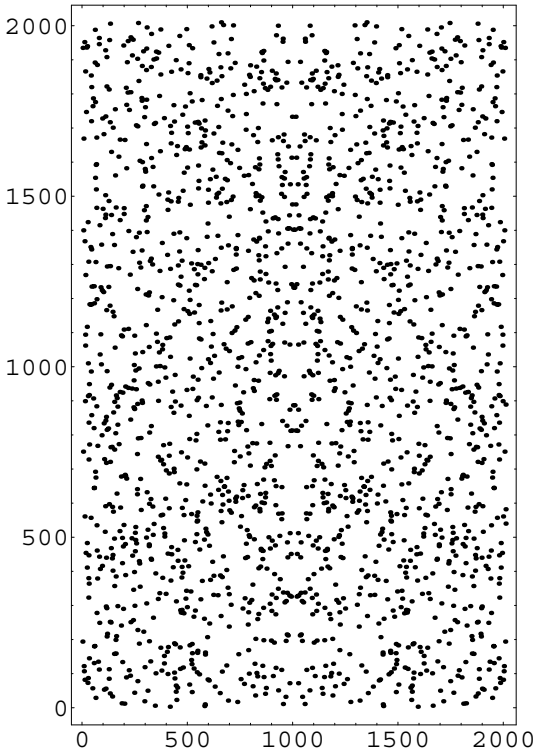
Consideremos el cuerpo de p elementos $\mathbb{F}_p = \{0, 1, \ldots, p-1\}$, con las operaciones de suma y multiplicación (mód p). Una curva elíptica $E(\mathbb{F}_p)$ sobre $\mathbb{F}_p$, donde $p \neq 2, 3$, es el conjunto de los puntos $(x, y) \in \mathbb{F}_p^2$ que satisfacen una ecuación de la forma:

$$E : y^2 = x^3 + ax + b$$

donde $a, b \in \mathbb{F}_p$ y $4a^3 + 27b^2 \neq 0$, junto con un punto adicional $\mathcal{O}$ llamado *punto del infinito*. $-(4a^3 + 27b^2)$ es el *discriminante* del polinomio cúbico $x^3 + ax + b$ que define la curva y la

condición de que no se anule es equivalente a que este polinomio no tenga ceros múltiples o, lo que es lo mismo, a que la curva definida por la ecuación anterior sea *no singular* o *lisa* (no existen puntos de la curva, considerada sobre una "clausura algebraica" de $\mathbb{F}_p$, en los que ambas derivadas parciales de la cúbica de su ecuación implícita se anulan simultáneamente).

Veamos el aspecto de una curva elíptica sobre un cuerpo finito, que es bastante diferente al de la gráfica habitual sobre $\mathbb{R}$. Consideremos la curva de ecuación $y^2 = x^3 + 566x + 657$ sobre $\mathbb{F}_{2011}$. En la gráfica, los ejes de las x y de las y están intercambiados con respecto a la representación habitual, para que la curva muestre con más claridad su "temible simetría":



La ecuación $y^2 = x^3 + ax + b$ se llama la *ecuación de Weierstrass* de la curva. La estructura de grupo abeliano de esta curva está dada por unas fórmulas, que son las que se manejan en los cálculos criptográficos, pero tiene una descripción geométrica muy sencilla: si una recta corta a la curva en dos puntos, entonces también la corta en un tercer punto único (aquí hay que tener en cuenta que los puntos de intersección hay que contarlos con su multiplicidad correspondiente y también que el punto del infinito pertenece a la curva). Pues bien, la suma de los dos puntos iniciales es el simétrico, con respecto al eje de las x, del tercer punto en que la recta que determinan corta a la curva. La suma está caracterizada por ser $\mathcal{O}$ el elemento neutro junto con el "principio de la secante-tangente": *la suma de tres puntos de la curva es $\mathcal{O}$ si y sólo si los tres puntos están alineados.* Por ejemplo, si queremos calcular la suma de un punto P consigo mismo (que, siguiendo la notación usual, se denota $2P$), habrá que trazar la tangente a la curva en ese punto (la tangente existe, por ser la curva no singular). La tangente tiene un contacto de orden ≥ 2 en P y cortará a la curva en un tercer punto (que puede ser el propio P si la multiplicidad de la intersección es 3, en cuyo

caso P es un "punto de inflexión" y, de acuerdo con la regla anterior, será $P+P=-P$ o, lo que es lo mismo, $3P=\mathcal{O}$; los puntos de inflexión distintos de $\mathcal{O}$ son precisamente los puntos de orden 3 en esta operación). Es fácil ver que la operación así definida le da estructura de grupo al conjunto de puntos de la curva; lo única propiedad no evidente es la asociatividad que se puede demostrar, bien analíticamente, o bien usando la teoría de intersecciones de curvas (véase [2, 26] para todo lo relacionado con la criptografía de curvas elípticas).

El criptosistema de El Gamal elíptico funciona de la forma siguiente. Alicia y Bernardo comparten una curva elíptica E y un punto $P \in E$. Además, cada uno de ellos tiene una clave privada, que es un entero comprendido entre 2 y $|E|-1$ (donde $|E|$ denota el orden de E), digamos a en el caso de Alicia y b en el caso de Bernardo. Sus correspondientes claves públicas son los puntos de la curva aP y bP (recuérdese que aP es el punto obtenido sumando P a veces en la curva). Supongamos entonces que Alicia quiere enviar un mensaje a Bernardo y que todos los textos claros están representados por puntos de la curva (lo cual no presenta dificultad alguna). Entonces, si Alicia quiere enviar a Bernardo $Q \in E$, en forma cifrada, le envía el par de puntos $(kP, Q+k(bP))$, donde k es un entero que ha elegido aleatoriamente. Bernardo multiplica el primero de estos puntos por su clave privada b, calculando $b(kP)=k(bP)$. A continuación, resta este punto del segundo punto que ha recibido, es decir, calcula $Q=Q+k(bP)-b(kP)$ y ya tiene el texto claro. En cambio, Eva, intenta calcular Q a partir del par $(kP, Q+k(bP))$ pero, como no conoce ni k ni b, no lo tiene fácil; aparentemente, tendría que ser capaz de resolver el PLDE sobre esta curva para poder calcular k o b y, a continuación Q (obsérvese que, puesto que estamos trabajando en un grupo aditivo, si $kP=R$, k es el logaritmo discreto de R en la base P y un ejemplo del PLDE es el cálculo de k dados P y R). Aquí es donde la ventaja de usar curvas elípticas en lugar de un grupo $\mathbb{Z}_p$ se pone de manifiesto, pues para resolver el PLDE sólo se conocen algoritmos de tiempo exponencial, lo que significa que, al menos por el momento, este problema es mucho más difícil que el PLD usual, que ya de por sí dista mucho de ser computacionalmente fácil.

Para que el PLDE sobre una curva elíptica concreta E resulte difícil, la curva y el punto P tienen que ser elegidos con cuidado y cumplir una serie de condiciones que no detallaré, aunque si mencionaré que el número de puntos de la curva (su orden) debe ser grande y, a de ser posible, primo. Si esta última condición se cumple, se estima que "grande" significa que debe tener al menos 160 bits, es decir, unos 48 dígitos decimales. Una condición imprescindible para poder implementar un criptosistema sobre una curva elíptica es la de poder calcular su orden. Por suerte, existen algoritmos eficientes (de tiempo polinómico) para este fin, en particular, el llamado *algoritmo de Schoof* y sus derivados.

Una curva elíptica de Microsoft

La importancia creciente de las curvas elípticas en criptografía se pone de manifiesto por el hecho de haber sido usadas por Microsoft en su software, concretamente, en la versión 2 del llamado "Microsoft Digital Rights Management" (MS-DRM), que se aplica a los derechos de reproducción de ficheros de audio en formato .wma (en concreto, a la versión 7 de "Windows Media Audio") y que ha sido descrito por "Beale Screamer" (en adelante, BS), un criptólogo anónimo que, en octubre de 2001 envió al grupo de noticias `sci.crypt` los detalles, junto con una forma de romper el sistema [1]. La curva elíptica de Microsoft, que denotaré M, está definida sobre el cuerpo $\mathbb{F}_p$, donde

$$p = 7859631023794288223766947894468973962074985689 51$$

Este número no es fácil de recordar pero, si se escribe en el sistema de numeración hexadecimal, es 89ABCDEF012345672718281831415926141424F7, lo que proporciona una regla mnemotécnica fácil. Si se excluyen los dígitos '4F7' del final, los restantes son: los 16 dígitos del sistema hexadecimal, en orden creciente cíclico empezando por el 8, los 8 primeros dígitos de e, los 8 primeros dígitos de π, y los cinco primeros dígitos de $\sqrt{2}$, lo que proporciona una regla mnemotécnica fácil (y un alto *"nerd appeal"*, en palabras de BS). Por otra parte, la ecuación de la curva, es M: $y^2 = x^3 + ax + b$, donde a y b son, dados también en hexadecimal:

a = 37A5ABCCD277BCE87632FF3D4780C009EBE41497;
b = 0DD8DABF725E2F3228E85F1AD78FDEDF9328239E

El orden de la curva puede calcularse usando el algoritmo de Schoof, y resulta ser n = 78596310237942882237669302488171495761268615742 9, el cual es, además, primo. Esto no es casualidad, sino que ha sido buscado por Microsoft para aumentar la dificultad del PLDE sobre esta curva. Se puede observar también que n es un números de 160 bits, de modo que tiene tamaño suficiente. Dado que el orden de la curva M es primo, cualquier punto es un generador del grupo y podría ser usado como base del logaritmo discreto. El punto elegido por Microsoft para este fin es el punto P de M de coordenadas hexadecimales:

P = (1623947FD6A3A1E53510C07DBA38DAF0109FA120,
16D5744911075522D8C3C5856D4ED7ACDA379936F)

Veamos ahora como se usa el criptosistema de El Gamal sobre M en el esquema MS-DRM. Como he indicado ya, si Bernardo es un usuario, tendría que elegir como clave privada un entero b tal que $1 < b < n - 1$ pero, en este caso, su clave privada la ha generado el software de Microsoft, que la ha ocultado en distintos ficheros del ordenador de Bernardo. Cuando Bernardo compra música a través de Internet, su ordenador se pone en contacto con el servidor de licencias del vendedor y le envía su clave pública bP. Entonces, el sistema del vendedor le envía a Bernardo un mensaje cifrado mediante el criptosistema de El Gamal elíptico, usando la clave pública de Bernardo; este mensaje es la licencia que permitirá a Bernardo reproducir en su ordenador los ficheros de audio que ha adquirido, que están, a su vez, cifrados mediante criptosistemas simétricos. El mensaje cifrado que el ordenador de Bernardo recibe es un par de puntos de la curva M, de la forma: $(kP, Q + k(bP))$, y la clave secreta necesaria para descifrar los ficheros de audio es la primera coordenada del punto Q. El ordenador de Bernardo usa la clave privada b para descifrar el mensaje en la forma antes vista y calcular $Q = Q + k(bP) - b(kP)$. La primera coordenada de Q permite a Windows Media Player descifrar y reproducir los ficheros de audio que están cifrados usando DES, RC4 y otros sistemas, en una forma bastante complicada que no voy a describir. Por lo tanto, Bernardo ya puede escuchar su música.

Pero supongamos que Bernardo quiere compartir esta música con su amiga Carolina. Para ello le pasa el fichero .wma correspondiente y también el fichero que contiene la licencia $(kP, Q + k(bP))$. Sin embargo, cuando Carolina quiere reproducir la música, no puede hacerlo, porque su software trata de recuperar Q haciendo el cálculo $Q + k(bP) - c(kP)$, donde c es la clave privada de Carolina (claro está que el ordenador de Carolina no conoce la clave privada b de Bernardo). Como $kbP \neq kcP$, el punto resultante no es Q y de ahí que Carolina no pueda reproducir la música.

En [1] se muestra la forma de romper el sistema de protección de los ficheros de audio .wma usado en MS-DRM, mediante la recuperación de la clave privada del usuario correspondiente al criptosistema de El Gamal elíptico, lo que permite a su vez recuperar, a partir de una licencia válida, la clave de contenido que permitirá reproducir este en cualquier sistema. Sin embargo, esto no significa que el criptosistema basado en curvas elípticas haya sido roto, pues no se conoce la forma de recuperar la clave secreta a partir de la clave pública de Bernardo y si Eva intercepta la licencia enviada a Bernardo, no puede usarla para reproducir el contenido. Lo que ocurre en este caso es que la clave privada de Bernardo está oculta en su propio ordenador y BS ha encontrado la forma de que Bernardo pueda recuperarla. La forma de proporcionar a cada usuario una clave privada que le permita reproducir la música que ha adquirido en su ordenador, sin permitirle que pueda transferir esa clave privada a otros ordenadores u otras personas es, por así decirlo, un problema de "criptografía dentro de la criptografía". La debilidad de la versión 2 del MS-DRM que permitía recuperar la clave privada ya ha sido corregida y, como indicaré más adelante, parece ser que Microsoft tiene intención de extender a otros ámbitos el uso de la criptografía elíptica.

El episodio anterior se puede ilustrar perfectamente con la siguiente tira cómica:

En el caso que nos ocupa, se podría decir que BS le ha robado la cartera a Bill Gates, pero este quizá no se preocupe demasiado porque tiene *muchas* carteras y, además, puede que la próxima vez ya no sea tan fácil ...

Epílogo

La matematización de la criptografía ha revolucionado esta en varios aspectos. Por una parte, ha permitido la aparición de la criptografía de clave pública, que es especialmente adecuada para su uso masivo a través de las redes informáticas. Por otra, ha permitido alcanzar un nivel de seguridad muy alto como se comprueba por el hecho de que las autoridades de algunos paises –y, en particular, las de EEUU– han impuesto serias limitaciones a la exportación de "criptografía fuerte". Durante el último cuarto del siglo XX, la criptografía era considerada en EEUU, a efectos de exportación, como una munición, y en algunos casos esto dio lugar a episodios curiosos como el que se produjo cuando la primera edición del

libro de Bruce Schneier *"Applied Cryptography"* fue considerada exportable pero no así el disquette que contenía el código fuente en C que también figuraba en el libro. Parece que el hecho de teclearlo en un ordenador para convertirlo en formato electrónico era lo que había convertido en "munición" a este código ... Finalmente, las restricciones a la exportación de criptografía fuerte fueron levantadas –salvo a media docena de países– por la administración Clinton en 2000, no sin antes haber intentado establecer un estándar criptográfico –el algoritmo *Skipjack* y el chip *Clipper*– que permitiría a las autoridades descifrar cualquier mensaje haciendo uso de unas claves depositadas en ciertas entidades, y que finalmente fue descartado ante la oposición de la sociedad y la de la industria del software.

Estos hechos avalan la creencia de que estamos en un momento histórico en el que los criptógrafos disfrutan de una posición muy fuerte ante los criptoanalistas, debido precisamente al uso de las herramientas matemáticas en criptografía. Uno de los medios más eficaces para evaluar la seguridad de los criptosistemas actuales consiste en la organización de concursos –como el ya mencionado *RSA Factoring Challenge*– en los que algunas entidades ofrecen premios a quienes sean capaces de resolver casos concretos de problemas computacionales difíciles como son la factorización de enteros y el PLDE. La idea es monitorizar el nivel de seguridad que ofrecen los criptosistemas basados en estos problemas controlando los avances que se producen en los algoritmos para resolverlos y en sus implementaciones. Las factorizaciones ya mencionadas de módulos de RSA de 512 bits (llamado RSA-155 porque tenía 155 dígitos decimales) y de 576 bits (RSA-576) muestran que los módulos de 512 bits, que se venían usando hasta hace poco en muchas implementaciones comerciales, ya no son seguros, y el tamaño mínimo del módulo que se recomienda actualmente es de 1024 bits. Con los mejores algoritmos actuales, la factorización de un módulo de RSA de 1024 bits requiere unos 10^{12} años MIPS, donde 1 año MIPS es la computación que puede realizar una máquina que procese un millón de instrucciones por segundo (1 MIPS), durante un año. Esto se puede comparar con las factorizaciones de RSA-129 y RSA-155 que requirieron, respectivamente, 5.000 y 8.000 años MIPS (hay que tener en cuenta que en el caso de RSA-129 se usó MPQS, que es un algoritmo inferior a NFS). Se estima que un nivel de seguridad equivalente al proporcionado por un módulo de RSA de 1024 bits (es decir, 10^{12} años MIPS) se alcanza, para los criptosistemas basados en el PLDE como El Gamal elíptico, con módulos de 160 bits, lo cual proporciona una gran ventaja en eficiencia a los sistemas basados en curvas elípticas. Existe también un concurso, con premios en metálico, para resolver el PLDE, la *Certicom Elliptic Curve Challenge* [3] y, por el momento, el módulo más grande para el que se ha conseguido resolver el PLDE, tiene 109 bits.

Una situación concreta en la que la ventaja de las curvas elípticas se pone de manifiesto, es mencionada en unas notas de William Stein, donde atribuye al criptógrafo de Stanford Dan Boneh la observación de que Microsoft va a usar pronto un criptosistema basado en curvas elípticas durante la instalación de su software. El motivo es que no es muy razonable pedir al usuario que teclee una clave muy larga, y las curvas elípticas permiten reducir sensiblemente su tamaño dentro de un margen de seguridad dado. Analicemos un poco esta cuestión y, a tal efecto, imaginemos que para la activación del software se le pide al usuario que introduzca una clave que no es otra cosa que un logaritmo discreto elíptico (un entero m tal que $mP = Q$ en una curva elíptica dada). El software conoce P, Q y la curva y comprueba –cosa que puede hacerse rápidamente– si $mP = Q$. En caso afirmativo, el software queda activado pero, si el valor de mP no coincide con Q, la activación es denegada. Obsérvese que el software no necesita conocer m (la situación es análoga a la que se produce en los cajeros automáticos, donde el PIN juega un papel similar al de m). El valor de m sería proporcionado al usuario al adquirir el programa, pero si alguien quiere

activar un programa pirateado y no conoce m, tendría que obtenerlo calculando el logarítmo discreto elíptico de Q en la base P, sobre la curva elíptica utilizada, lo cual significaría ser capaz de resolver un PLDE. Como ya he indicado, el PLDE con un módulo de 160 bits proporciona un nivel de seguridad equivalente al de RSA con una clave de 1024 bits, que se considera razonable en la actualidad. Si se adoptase un módulo de esta longitud, el tamaño de m no sería superior a 160 bits. Si se utilizan los 26 caracteres alfabéticos del idioma inglés, distinguiendo mayúsculas de minúsculas y usando también los 10 dígitos decimales, es decir, utilizando el sistema de numeración de base 62, un número de 160 bits se puede escribir con no más de $1 + \lfloor 160 \log_{62} 2 \rfloor = 27$ caracteres, que se podrían introducir (con un poco de trabajo) a través del teclado. Sin embargo, si se usase RSA para el mismo propósito, el usuario tendría que teclear el equivalente a 1024 bits, lo que podría representar hasta 172 caracteres alfanuméricos, algo que ya no parece razonable.

Una última observación en relación con la seguridad de los criptosistemas de clave pública es que hay que ser precavidos al valorar los resultados de los concursos antes mencionados y no se puede excluir que alguien pueda, por ejemplo, descubrir un método de factorización eficiente sin hacerlo público, lo que le permitiría descifrar fácilmente toda la información cifrada con RSA. Aunque ello parece poco probable en la actualidad, no hay que olvidar que hay organizaciones poderosas para las cuales esto sería sin duda un objetivo muy deseable. El ejemplo mejor es la NSA, que dedica enormes recursos a descifrar todo tipo de información cifrada.

La existencia de la NSA fue mantenida oficialmente en secreto durante muchos años,

hasta el punto de que, en los círculos que sabían de ella, se la denominaba, haciendo un juego de palabras con sus iniciales, "No Such Agency" ("No hay tal agencia"). Actualmente, es la organización de todo el mundo que emplea a un mayor número de matemáticos, superando a cualquier Universidad o Centro de Investigación y seguramente Lenstra y Verheul estaban pensando en ella cuando escribieron en [15] refiriéndose a RSA155: *"... una clave de RSA de* 512 *bits fue factorizada en agosto de 1999. Aunque este resultado es la primera factorización publicada de un módulo RSA de* 512 *bits, sería ingenuo creer que esta es la primera vez que una tal factorización ha sido obtenida."*.

Por otra parte, ya he mencionado que la posible llegada de los ordenadores cuánticos sería letal para RSA, aunque no necesariamente para la criptografía de clave pública en general. Además, los ordenadores cuánticos permitirían, utilizando el hecho de que un canal cuántico *no puede ser escuchado pasivamente*, realizar intercambios de claves con seguridad total, pues dichas claves no podrían ser obtenidas subrepticiamente por un criptoanalista, ni siquiera en el caso de que dispusiese de un ordenador cuántico. Tampoco está del todo claro si el intercambio de claves cuántico será pronto una realidad práctica, pero parece mucho más factible que los ordenadores cuánticos y, a nivel experimental, los resultados son alentadores pues se han realizado intercambios de claves, aunque a distancias reducidas.

Para terminar, haré un comentario sobre el hecho de que las matemáticas resulten tan útiles en criptografía. Este es un ejemplo más de lo que se podría denominar la *irrazonable efectividad de las matemáticas*, parafraseando al premio Nobel Eugene Wigner, quien escribió [27]:

> *... la enorme utilidad de las matemáticas en las ciencias naturales es algo que bordea lo misterioso y ... no existe una explicación racional de ello.*

Wigner estaba pensando fundamentalmente en la física, pero algo similar se observa en la criptografía, donde conceptos y resultados matemáticos muy anteriores –los primos y su distribución, la factorización de enteros, las curvas elípticas–, que se habían desarrollado sin pensar en la posibilidad de aplicación alguna, han resultado ser decisivos para el avance de la criptografía. Este hecho es aun más sorprendente si se tiene en cuenta que era algo inimaginable en tiempos bastantes recientes como se pone de manifiesto en lo que escribió G.H. Hardy, en 1940, su libro autobiográfico *Apología de un matemático* [11], cuando declaró que la teoría de números no tenía aplicaciones, aunque justo es reconocer que la mayor parte de ellas eran difíciles de imaginar en aquella época.

Resulta sorprendente, por ejemplo, que no dejen de descubrirse nuevas aplicaciones de las curvas elípticas a la criptografía, utilizando cada vez resultados más profundos, los cuales habían surgido con motivaciones totalmente diferentes. Para ilustrar esto mencionaré brevemente una de estas nuevas aplicaciones criptográficas, que ha experimentado un notable desarrollo a partir del año 2001. Su origen se remonta a la publicación, en 1985, de un artículo de Adi Shamir, en el que proponía un nuevo tipo de criptografía que ha dado en llamarse *"criptografía basada en la identidad"* ("Identity-based Cryptography" o, más brevemente, ID-based Crypto, o ID-C). En un criptosistema de clave pública usual (CCP), cada usuario tiene una clave pública y una clave privada, donde la primera se genera (por el propio usuario o por una "autoridad central") aplicando una función unidireccional a la segunda. El principal problema cuando se usa un CCP es el de la autenticidad de la clave pública. La clave pública de Bernardo es todo lo que Alicia necesita conocer para enviarle un mensaje cifrado, pero si Eva consigue convencer a Alicia de que una clave que la propia Eva ha construido es la clave pública de Bernardo, Eva será capaz de descifrar los mensajes que

Alicia le envíe a Bernardo creyendo que está usando la clave pública de este. Esto hace que sea imprescindible que los usuarios de un CCP puedan verificar la autenticidad de las claves públicas de los restantes usuarios. La solución habitual a este problema es el uso de lo que se llama una *infraestructura de clave pública* ("public-key infrastructure", PKI), que suele incluir una *autoridad certificadora* ("Certification Authority", CA), que es quien garantiza la autenticidad de una clave pública mediante un certificado firmado con su propia clave secreta. La idea de Shamir fue evitar el uso de la autentificación (y, por tanto, de la PKI), haciendo que la clave pública de cada usuario esté intrínsecamente ligada a su identidad y pueda, en consecuencia, deducirse directamente de la información pública que lo identifica, llamada su *identidad digital*, y que puede incluir datos como el nombre, el NIF, o la dirección de correo electrónico. Dado que la clave pública ha de estar determinada por la identidad del usuario, no se puede generar a partir de la clave privada como en un CCP y, en consecuencia, lo que se hace es seguir exactamente el camino inverso, es decir, la clave privada se genera a partir de la clave pública. Sin embargo, esto plantea un problema. Si la generación de claves fuese llevada a cabo por los propios usuarios, entonces cada uno de ellos sabría como deducir su clave privada a partir de su clave pública y, por el mismo método, podría deducir las claves privadas de otros usuarios a partir de sus claves públicas. Por tanto, las claves deben ser generadas por una *autoridad de confianza* ("Trusted Authority", TA), que distribuirá las claves privadas a los usuarios, las cuales habrán sido obtenidas mediante una función de la forma:

```
clave privada = F(clave maestra, clave pública)
```

donde la *clave maestra* es posesión exclusiva de la TA.

En el mismo artículo en el que definió la ID-C, Shamir ya propuso una implementación de un protocolo de firmas digitales basado en ID-C, pero aun iba a pasar bastante tiempo antes de que se pudiese obtener un criptosistema de este tipo (el propio Shamir observó que era imposible convertir RSA en un criptosistema basado en la identidad). La solución iba a venir de mano de las curvas elípticas y, más concretamente, de ciertas aplicaciones bilineales llamadas *pareos* ("pairings") definidos sobre subgrupos de ciertas curvas elípticas. Se conocen dos pareos no triviales que tienen las propiedades requeridas, el pareo de Weil y el pareo de Tate. El pareo de Weil ya era muy importante en la teoría de las curvas elípticas antes de descubrirse conexión alguna con la criptografía. A grandes rasgos, la razón es que permite obtener "información global" a partir de "información local" y se usa, por ejemplo, en la demostración del *teorema de Hasse*, un resultado muy importante que acota el número de puntos de una curva elíptica sobre un cuerpo finito [26]. Curiosamente, su primer uso criptológico no fue en criptografía sino en criptoanálisis y se produjo cuando Menezes, Okamoto y Vanstone descubrieron un ataque (llamado el "ataque MOV" o la "reducción MOV") que reduce el PLDE sobre ciertas curvas elípticas (en particular, sobre las llamadas *supersingulares*) al PLD sobre el grupo multiplicativo de un cuerpo finito. La ventaja es que, mientras que para el PLDE en general, solo se conocen algoritmos exponenciales, para el PLD sobre el grupo multiplicativo de un cuerpo finito existen algoritmos subexponenciales que son mucho más eficientes, por lo que las curvas supersingulares no deben ser usadas en criptosistemas como el de El Gamal elíptico. Un año después de descubrirse la reducción MOV, Frey y Rück descubrieron la llamada *reducción FR*, que es similar pero utiliza el pareo de Tate en lugar del de Weil. Dado que el pareo de Tate se puede computar más eficientemente que el de Weil, la reducción FR es superior a la reducción MOV. Sobre curvas elípticas arbitrarias, ninguna de las dos reducciones es efectiva debido a la dificultad

de computar los pareos en general.

Se dice a menudo que la aplicación de los pareos que acabo de mencionar es *destructiva* –obviamente lo es desde el punto de vista del criptógrafo, aunque no así desde el del criptoanalista–. Curiosamente, después de las aplicaciones destructivas llegaron las constructivas pues, como ya he indicado, los pareos son los instrumentos que, finalmente, permitieron la realización plena de la criptografía basada en la identidad que Shamir había propuesto y, más concretamente, permitieron construir criptosistemas de *cifrado basado en la identidad* ("Identitity-Based Encryption", IBE). El primero de estos criptosistemas fue descubierto por Boneh y Franklin en 2001 y su implementación concreta se puede basar tanto en el pareo de Weil como en el de Tate, siendo este último el que proporciona mayor eficiencia. Para que los pareos sean eficientemente calculables, hay que usar curvas supersingulares (es precisamente esta característica la que las hace vulnerables a los ataques MOV y FR en la criptografía elíptica usual), o bien otras curvas construidas especialmente. Está fuera del alcance de estas notas dar una descripción del sistema de Boneh-Franklin, que ya empieza a aparecer en los textos y se está descrito, por ejemplo, en [26, 16]. También está descrito en la página web sobre criptografía de la Universidad de Stanford [12] donde, además del artículo original de Boneh y Franklin, se puede obtener el código fuente de una aplicación de correo electrónico seguro basado en IBE que han desarrollado.

El esquema IBE de Boneh-Franklin no es, ni mucho menos, la única aplicación criptográfica de los pareos sobre curvas elípticas; existen muchas otras de diversos tipos y un buen sitio para aprender sobre ello es la página de Paulo Barreto [17]. Volviendo a la efectividad de las matemáticas en criptografía, hay que reconocer que es realmente misterioso, por no decir inexplicable, que el pareo de Weil sirva para cosas tan diferentes como demostrar el teorema de Hasse, atacar el criptosistema de El Gamal elíptico o, por el contrario, realizar un criptosistema basado en la identidad. Estos son sólo una parte de los misterios que encierran las curvas elípticas, que cada vez tienen un papel más preponderante en las matemáticas de nuestro tiempo. Pero esa es otra historia ...

Referencias

[1] "Beale Screamer", Microsoft's Digital Rights Management Scheme - Technical Details, en: `http://cryptome.org/beale-sci-crypt.htm`.

[2] I. Blake, G. Seroussi, N. Smart, *Elliptic Curves in Cryptography*, Cambridge University Press (1999).

[3] Certicom Elliptic Curve Challenge, en: `http://www.certicom.ca/index.php?action=res,ecc_challenge`.

[4] R. Crandall, C. Pomerance, *Prime numbers, A computational perspective*, Springer-Verlag (2001).

[5] Cryptanalysis of block ciphers with overdefined systems of equations, Asiacrypt '02, Lecture Notes in Computer Science 2501, pp. 267-287, Springer-Verlag (2002).

[6] W. Diffie and M. Hellman, New directions in cryptography, IEEE Transactions on Information Theory 22 (1976), 644-654.

[7] J.L. Gómez Pardo, Aspectos computacionales de los números primos (II), La Gaceta de la RSME 5, no. 1 (2002), 197-227.

[8] J.L. Gómez Pardo, Criptografía y curvas elípticas, La Gaceta de la RSME, Vol. 5.3 (2002), 737-777.

[9] J.L. Gómez Pardo, The Advanced Encryption Standard, en: `http://library.wolfram.com/infocenter/MathSource/5130/`

[10] Project Gutenberg, en: `http://www.gutenberg.net/etext/2000`.

[11] G.H. Hardy, *Apología de un matemático*, Ariel (1981).

[12] Identity-Based Encryption, en: `http://crypto.stanford.edu/ibe/`.

[13] Neal Koblitz, *Algebraic Aspects of Cryptography*, Springer-Verlag (1998).

[14] H.W. Lenstra, Jr., Rijndael for algebraists, en: `http://math.berkeley.edu/~hwl/`.

[15] A.K. Lenstra, E.R. Verheul, Selecting cryptographic key sizes, J. Cryptology 14 (2001), 255-293.

[16] Wenbo Mao, *Modern Cryptography, Theory & Practice*, Prentice Hall (2004).

[17] The Pairing-Based Crypto Lounge, en: `http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html`.

[18] E. A. Poe, El escarabajo de oro, en Narraciones Completas, Aguilar (1964).

[19] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public key cryptosystems, Comm. ACM 21 (1978), 120-126.

[20] The New RSA Factoring Challenge, en: `http://www.rsasecurity.com/rsalabs/node.asp?id=2092`.

[21] C.E. Shannon, Communication theory of secrecy systems, Bell Systems Technical Journal 28 (1949), 656-715.

[22] Peter W. Shor, Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Computing 26 (1997), 1484-1509.

[23] Simon Singh, *The Code Book*, Fourth Estate, London (1999).

[24] D. Stinson, *Cryptography, Theory and Practice*, 2nd. Ed., Chapman & Hall/CRC (2002).

[25] W. Trappe, L.C. Washington, *Introduction to Cryptography with Coding Theory*, Prentice Hall (2002).

[26] L.C. Washington, *Elliptic curves*, Chapman & Hall/CRC (2003).

[27] E.P. Wigner, The unreasonable effectiveness of mathematics in the natural sciences, Comm. Pure Appl. Math. 13 (1960), 1-14.