[VOLUMEN 20 · JUNIO 2000]
Ágora SiC Divulgación

SUMARIO

Ataques a DES y módulos factorizados de RSA

Raúl Durán Díaz / ALCATEL
Luis Hernández Encinas / UNIVERSIDAD DE SALAMANCA
Jaime Muñoz Masqué
Dpto. de Tratamiento de la Información y Codificación
Instituto de Física Aplicada, CSIC

En este artículo se presentan los últimos ataques a DES y las factorizaciones de cuatro módulos de RSA, los dos criptosistemas más utilizados en la actualidad. Se comentan los éxitos en el criptoanálisis de las claves de DES y la necesidad de usar otros criptosistemas de clave secreta con longitudes de clave mayores. Se presentan también las factorizaciones de diferentes módulos de RSA. En este último caso se analizan, de forma elemental, los métodos matemáticos y computacionales utilizados, se pone de manifiesto la robustez de este criptosistema y se indican las longitudes de las claves de RSA recomendables para los próximos años.

Ataques a DES y módulos factorizados de RSA

INTRODUCCIÓN

Es sabido que hasta la fecha no existe prueba alguna que garantice la absoluta seguridad de los algoritmos de cifrado implementados y utilizados habitualmente en diferentes ámbitos: militar, diplomático, correo electrónico, pagos electrónicos, etc. No se puede, pues, garantizar absolutamente que informaciones confidenciales o reservadas no lleguen a ser del dominio de un tercero no autorizado. Como ejemplo doméstico, podemos imaginar el peligro que puede acarrear la transmisión por Internet del número de nuestra tarjeta de crédito o de nuestra cuenta bancaria para realizar un pago, si este dato es interceptado por una tercera persona. Aunque hay recursos para anular un pago que se presume fraudulento, es más prudente no correr riesgos. Por tanto, resulta de vital importancia conocer los avances en los desarrollos teóricos y tecnológicos que puedan llevar a romper las implementaciones de los criptosistemas más comunes. Este conocimiento, si bien no garantiza el perfecto secreto de la información transmitida, podría permitir al menos estimar el grado de confianza que el sistema utilizado nos proporciona. Desde el punto de vista de un usuario, no se trata de saber si tal problema de Teoría de Números ha sido resuelto o si tal estimación de Complejidad Computacional ha sido mejorada. Se trata de conocer hasta dónde se ha llegado en el criptoanálisis, para tener una idea de la fiabilidad del sistema empleado.

Este artículo se estructura como sigue. En primer lugar se presentan los resultados obtenidos sobre la seguridad de DES, que ponen de manifiesto la necesidad de utilizar otros criptosistemas de clave secreta con longitudes de clave mayores. A continuación se explica el funcionamiento del criptosistema RSA para, más adelante, analizar su seguridad por medio del algoritmo de la criba cuadrática, utilizado para la factorización de un número de 129 dígitos. Posteriormente se extiende el método anterior al de la criba del cuerpo de números y su uso para factorizar un módulo de RSA de 130 dígitos. En el siguiente apartado se incluyen nuevos módulos RSA factorizados, de 140 y 155 dígitos, que constituyen los récords hasta la fecha y, finalmente, se comentan las características de un nuevo dispositivo, el Twinkle, que ha sido propuesto para mejorar los tiempos de cálculo de factorizaciones. Por último, se apuntan algunas conclusiones a la vista de los resultados presentados a lo largo de este artículo.

SEGURIDAD DEL CRIPTOSISTEMA SIMÉTRICO DES

DES (Data Encryption Standard) es uno de los sistemas de clave secreta más utilizados. Emplea el llamado «cifrado en bloque», que descompone el mensaje original en bloques de igual tamaño, a cada uno de los cuales aplica un mecanismo idéntico de cifra. En concreto, DES es una particularización del cifrado de Feistel, en que se divide cada bloque en dos mitades y después se elige una operación no lineal que actúa repetidamente sobre cada una de ellas de forma alternada. Los bloques de DES son de 64 bits (ocho símbolos en codificación ASCII) y la longitud de la clave es de 56 bits. El espacio de claves de este criptosistema es, pues,

$$2^{56} = 72057\ 59403\ 79279\ 36 \approx 7{,}2\cdot 10^{16}.$$

Dado que a lo largo de este artículo se van a presentar números muy grandes –de más de 100 dígitos– se ha optado por seguir la norma habitual de separar sus dígitos en grupos de 5, comenzando por la izquierda, de modo que sea fácil contarlos.

Una interesante propiedad de DES es que la operación no lineal elegida es involutiva. Esto significa que la clave que se utiliza para cifrar sirve también para descifrar. Por esta razón, este sistema –como otros análogos– recibe el calificativo de simétrico.

DES fue el resultado del trabajo conjunto de la compañía IBM junto con la NSA (National Security Agency) de EEUU y vio la luz en la segunda mitad de los años setenta. Puesto que la NSA no quiso revelar algunos detalles acerca de la operación no lineal elegida, no falta quien piensa que quizá exista alguna «clave maestra», sólo conocida por la Agencia, que permita descifrar mensajes cifrados con cualquier otra clave. Fuera de esta sospecha, nunca confirmada, DES sólo ha sufrido a lo largo de su ya dilatada historia tres ataques realmente importantes.

El primero de ellos se llamó criptoanálisis diferencial y fue propuesto por Biham y Shamir (véase [BiSh91]). Muy resumidamente, consiste en elegir un gran número de textos en claro tales que la diferencia entre cada uno de ellos sea fija y conocida. A continuación se estudian las parejas texto claro-texto cifrado y se analiza cómo esa diferencia se va propagando a lo largo del algoritmo. Con esta técnica, el número de claves posibles se «reduce» a 2^{47}. Biham y Shamir experimentaron diversas variaciones del algoritmo DES y encontraron en todo caso que éste se debilitaba frente al análisis diferencial. Ello no era casualidad, pues Don Coppersmith, un investigador de IBM que trabajó en el diseño de DES, confirmó que éste se hizo teniendo en cuenta precisamente el análisis diferencial (véase [Lan00]).

El segundo ataque se llama criptoanálisis lineal, propuesto por Matsui [Mat93] . La idea es relativamente simple y hasta naïf: supóngase que el texto cifrado es una función lineal del texto en claro y de la clave. Inicialmente, ningún criptoanalista estaría dispuesto a admitir esto, pero Matsui fue capaz de encontrar funciones lineales que con altas probabilidades se satisfacían para sucesivas vueltas de DES. Finalmente llegó a la conclusión de que «bastaba» probar 2^{43} claves en promedio para hallar la verdadera. En 1994, una red de doce estaciones de trabajo fue capaz, usando el análisis lineal, de descifrar un mensaje cifrado con DES trabajando durante cincuenta días.

Llegamos así al sencillo pero temible ataque por fuerza bruta, que consiste simplemente en probar todas las claves posibles. Hasta hace dos años, este ataque por fuerza bruta contra DES era inviable debido al coste económico que suponía desarrollar una máquina capaz de probar todas las claves. Pero el imparable avance de la tecnología microelectrónica permitió a la EFF (Electronic Frontier Foundation) construir en mayo de 1998 una máquina, llamada DESCracker que, con un coste menor de 250.000 dólares, es capaz de probar todas las claves de DES en 9 días con lo que el tiempo medio para localizar una clave es tan solo 4,5 días (véase http://www.eff.org/DESCracker). Es de agradecer la honradez de la EFF que, conociendo y siendo capaz de romper DES, haya hecho públicos sus resultados y toda la información necesaria para la construcción de esta máquina en [EFF98]. Esta referencia sólo está disponible en papel y no se puede obtener en

formato digital debido a las restricciones del gobierno de EEUU en materia de exportación de tecnologías sobre Criptografía y Seguridad. Con ella, toda la comunidad científica ha podido saber que se trata de un conjunto de 36.864 unidades de pruebas de clave, cada una de la cuales prueba 2.500.000 de claves por segundo, lo que totaliza $9{,}216 \cdot 10^{10}$ claves por segundo. La máquina está gobernada por un PC compatible y está ensamblada en un bastidor que contiene dos chasis iguales. Cada chasis aloja 12 tarjetas, dotadas cada una de 64 chips cada uno de los cuales monta 24 unidades de pruebas de clave.

Ya antes del desarrollo de esta máquina, los Laboratorios RSA (http://www.rsa.com) trataron de demostrar la debilidad de DES frente a un ataque masivo y lanzaron en enero de 1997, el DES Challenge I (http://www.rsa.com/rsalabs/). Este desafío lo ganó Distributed.Net (una coalición mundial de unos 100.000 usuarios liderada por Rocke Verser), que consiguió encontrar la clave propuesta en 96 días. Más tarde, en febrero de 1998, un nuevo desafío, el DES Challenge II-1, fue de nuevo ganado por la Distributed.Net después de 41 días de trabajo. Posteriormente, el 17 de julio de 1998, EFF anunció que había logrado resolver el desafío de los Laboratorios RSA, DES Challenge II-2 (y ganado los 10.000 dólares de premio), en menos de 3 días (en 56 horas exactamente; véase, por ejemplo, [Lan00]).

Pues bien, fue la DESCracker de EFF, junto con Distributed.Net, quien volvió a romper un nuevo desafío lanzado por los Laboratorios RSA, el DES Challenge III (http://www.rsa.com/rsalabs/des3/index.html), encontrando la clave propuesta en tan sólo 22 horas y 15 minutos (y ganando de nuevo la recompensa ofrecida por estos laboratorios). Esta clave resultó de cifrar el mensaje «See you in Rome» («Nos vemos en Roma»), lugar de celebración de la Second Advanced Encryption Standard -AES- Conference, en marzo de 1999.

Aunque el coste de DesCracker no es accesible para un particular, sí lo es para un gobierno o una gran corporación, con lo que, a la vista de los resultados antes resumidos, se puede afirmar que DES ya no es un sistema seguro. Es claramente recomendable la utilización de criptosistemas simétricos con espacios de claves mayores, por ejemplo Triple DES o IDEA (véanse [FGHMM97], [MOV97] y [Sch96]).

CLAVE PÚBLICA: RSA

La seguridad del criptosistema de clave pública RSA ([RSA78]) se basa en la dificultad de factorizar un número entero ([FGHMM97], [MOV97] y [Sch96]).

El protocolo de uso del criptosistema RSA consta de tres pasos:

1. Generación de las claves
   Cada usuario U lleva a cabo las siguientes acciones:
   1.1. Elige dos primos grandes, distintos y de, aproximadamente, el mismo tamaño, p, q, y calcula $n = pq$.
   1.2. Selecciona un entero, e, primo con el orden de Z_n^*, es decir $1 < e < \varphi(n) = (p-1)(q-1)$ y $\mathrm{mcd}(e, \varphi(n)) = 1$.
   1.3. Calcula el inverso de e en Z_n^*, d: $ed \equiv 1 \pmod{\varphi(n)}$.
   1.4. Considera como su clave pública a la pareja (n, e), y como su clave privada a los números: (p, q, d, $\varphi(n)$).
2. Cifrado del mensaje m con la clave pública (n, e)
   Para cifrar un mensaje m, el remitente, B, ejecuta las acciones siguientes:
   2.1. Localiza la clave pública del destinatario A: (n, e).
   2.2. Calcula $m^e \pmod n = c$.
   2.3. Envía al destinatario el criptograma c.
3. Descifrado del criptograma c con la clave privada (d)
   Para recuperar el mensaje que se le ha enviado, el destinatario A
   3.1. Utiliza su clave privada d, para calcular $c^d \pmod n \equiv (m^e)^d \pmod n = m$.

Es fácil ver que si se determinan los factores del módulo RSA, n, el criptosistema RSA queda roto. En efecto, si $n = pq$, se calcula el valor de $\varphi(n) = (p-1)(q-1)$; con él y mediante el algoritmo de Euclides extendido, se determina el inverso d de e en Z_n^*, que es conocido por formar parte de la clave pública. Sabidos d y el criptograma, basta con llevar a cabo el paso 3.1. anterior para recuperar el mensaje original. La seguridad de este criptosistema se basa, pues, en que la factorización del módulo RSA, n, sea tan difícil (computacionalmente hablando) como sea posible.

Se sabe que factorizar el módulo RSA permite romper RSA; curiosamente, no está tan claro si existe otro método capaz de romper RSA, sin necesidad de factorizar n.

EL RSA-129 Y LA CRIBA CUADRÁTICA

Rivest, uno de los coautores de RSA, propuso en 1977 que se intentara factorizar un número de 129 dígitos con el fin de probar la robustez del sistema ([Riv77]). Él vaticinó que harían falta alrededor de $4 \cdot 10^{16}$ años de computación para lograrlo, según el estado de la Teoría de Números y la disponibilidad de algoritmos de entonces. Sin embargo, la factorización del número propuesto por Rivest, el famoso RSA-129, se logró el 2 de abril de 1994, después de menos de 8 meses de trabajo, merced al desarrollo del método conocido como la criba cuadrática (QS, Quadratic Sieve) descrito por Pomerance en [Pom85].

La iniciativa partió de la Universidad de Syracuse, NY, donde un grupo de investigadores propuso repartir en Internet el trabajo entre miles de voluntarios (véase http://www.npar.syr.edu/factoring/overview/CallToArms.txt). Cada voluntario recibió un programa que aprovechaba los momentos ociosos de su CPU. Después de determinados cálculos, el ordenador o el propio usuario devolvía sus resultados parciales a la sede central, que los almacenaba para su posterior uso. Después de varios meses de computación en paralelo, consiguieron factorizar el número RSA-129:

RSA-129 = 11438 16257 57888 86766 92357 79976 14661 20102 18296
72124 23625 62561 84293 57069 35245 73389 78305 97123
56395 87050 58989 07514 75992 90026 87954 3541
= 34905 29510 84765 09491 47849 61990 38981 33417 76463 84933
87843 99082 0577 · 32769 13299 32667 09549 96198 81908 34461
41317 76429 67992 94253 97982 88533,

lo que permitió descifrar el mensaje original de Rivest ([AGLL95]).

La criba cuadrática es un caso especial dentro de la familia de los métodos cuadráticos de factorización y consta de dos partes. La primera es la fase de criba, en que se calculan determinados residuos cuadráticos (un número a es un residuo cuadrático módulo n si existe otro número b tal que $a \equiv b^2 \pmod n$); la segunda es la fase de reducción de la matriz formada por los residuos cuadráticos obtenidos. En los siguientes párrafos vamos a dar una descripción más técnica del método; el lector menos interesado en estos detalles puede pasar directamente al siguiente apartado.

La idea básica de la familia de algoritmos cuadráticos es la siguiente. Siendo n el entero a factorizar, supongamos que x, y son dos enteros tales que $x^2 \equiv y^2 \pmod n$ pero $x \neq \pm y \pmod n$. Es claro entonces que $x^2 - y^2 = (x+y)(x-y) \equiv 0 \pmod n$, por lo que, si $x - y < n$, resulta que $\mathrm{mcd}(x-y, n) \neq 1$ ha de ser un factor de n. Esta idea es antigua y según [Pom85], que lo cita, ya fue publicada por Kraitchik en 1926 ([Kra26]).

A modo de ejemplo, tratemos de factorizar con este método el número 91. Después de algunos ensayos, nos daremos cuenta de que $10^2 \equiv 9 \pmod{91}$ y $3^2 \equiv 9 \pmod{91}$; por otro lado, $10 \equiv 10 \pmod{91}$ y $3 \equiv 3 \pmod{91}$, luego $x = 10$ e $y = 3$. Como $10 - 3 < 91$, tomamos $\mathrm{mcd}(10-3, 91) = 7$, que es un factor de 91.

Naturalmente el problema ahora es cómo encontrar de forma eficiente una pareja de números que cumpla esas condiciones. Vamos a tratar de solucionarlo en dos pasos. En primer lugar, elijamos un conjunto formado por los t primeros primos $F = \{p_1, p_2, ..., p_t\}$ al que denominamos base de factores. A cualquier entero que se factorice totalmente usando esta base de factores lo llamaremos uniforme de cota p_t o, simplemente, p_t-uniforme. Busquemos ahora parejas de enteros (a_i, b_i) tales que:

(i) $a_i^2 \equiv b_i \pmod n$,
(ii) b_i es p_t-uniforme.

En segundo lugar, obtenido un número suficiente de parejas, busquemos un subconjunto de entre los enteros b_i tales que su producto sea un cuadrado perfecto. Como tenemos la factorización de cada b_i en la base de factores, bastaría elegirlos de manera que las potencias de los primos p_j de la base de factores F en que se factoriza cada b_i sean pares; este problema equivale a encontrar la solución de un sistema lineal con coeficientes en Z_2, por lo que esta fase se suele llamar reducción de la matriz. Claramente el producto de sus correspondientes parejas a_i también será cuadrado perfecto -pues cada una lo es- luego hemos conseguido dos números tales que $x^2 \equiv y^2 \pmod n$. Si, además, $x \neq \pm y \pmod n$, bastará tomar $\mathrm{mcd}(x-y, n)$ y tendremos el factor buscado.

Este método admite mejoras y una de ellas conduce a la criba cuadrática. Consideremos el polinomio $f(x) = (x+m)^2 - n$, siendo m la parte entera de la raíz cuadrada de n. Se tiene que $f(x) = x^2 + 2mx + m^2 - n \approx x^2 + 2mx$, cuyo valor es pequeño con relación al valor de n, cuando el valor de x es pequeño en valor absoluto. Es fácil ver que $f(x) \equiv (x+m)^2 \pmod n$, con lo que ahora se satisface automáticamente la condición (i) anterior si $a_i = x_i + m$ y $b_i = f(x_i)$. Queda comprobar la condición (ii), pero con la ventaja de que ahora es más fácil determinar si b_i es p_t-uniforme, por ser un número más pequeño en valor absoluto.

Para determinar la uniformidad de b_i se puede utilizar el sencillo método de ensayar divisiones sobre la base de factores, pero justamente entra aquí la mejora introducida por la criba cuadrática. Para comenzar, observemos que si p es un primo de la base F que divide a f(x), un pequeño cálculo nos dirá que también divide a f(x+kp) para cualquier entero k. Resolvamos entonces la ecuación $f(x) \equiv 0 \pmod p$ y supongamos que, en el caso más general, las soluciones son x_1 y x_2. Obtendremos así dos series de valores $y_1 = x_1 + kp$, $y_2 = x_2 + kp$ que

son también solución de la ecuación y están equiespaciadas.

Estamos ya en disposición de comenzar la criba. Definamos un vector Q[x] con $-M \le x \le M$, cuya x-ésima entrada está inicializada a la parte entera de log |f(x)|. Elijamos un primo p de la base de factores F y supongamos que x_1, x_2 son las soluciones de la ecuación $f(x) \equiv 0 \pmod p$. A continuación, restamos el valor entero de log p de todas las componentes del vector Q tales que su índice $x \equiv x_1$ o $x_2 \pmod p$ y, naturalmente, $-M \le x \le M$; recordemos que todas esos valores de x son también solución de la ecuación $f(x) \equiv 0 \pmod p$. Repetimos esta operación para cada primo p de la base de factores. Después de la criba, las componentes del vector Q más próximas a cero son las candidatas a ser p_t-uniformes, porque su logaritmo era aproximadamente la suma de los logaritmos de los primos de la base de factores. Con esto se reduce el número de ensayos de divisiones sobre la base de factores a esos candidatos, lo que reduce a un mínimo el gasto computacional.

Para conseguir un número suficiente de parejas (a_i, b_i) sería necesario cribar un intervalo muy grande y, además, de su definición, se ve que |f(x)| crece linealmente con |x| con lo que decrece la probabilidad de ser p_t-uniforme. Para evitar este problema, se usa una variante del método, conocida como criba cuadrática con polinomios múltiples (MPQS, Multiple Polynomial Quadratic Sieve, [Sil87] , [Ley94]). En este caso, en vez de usar un único polinomio f(x) se usan muchos, adecuadamente elegidos, y se puede reducir la criba de cada polinomio a un intervalo mucho más pequeño.

De hecho, éste es el método que se usa en la práctica ya que es muy apropiado para la paralelización. Cada computador voluntario de la red recibe un programa y una asignación para cribar con una colección determinada de polinomios. Cuando encuentra un par satisfactorio, lo envía a la sede central, que acumula la colección de parejas que le llegan de los distintos nodos.

Cuando el número de parejas es suficiente se da por concluida la fase de criba. Tal como se explicó arriba, se trata ahora de encontrar un subconjunto $b_1, b_2, \ldots, b_m$, de los elementos b_i, tal que en la factorización de su producto, $b_1 \cdot b_2 \cdot \ldots \cdot b_m$, las potencias de cada uno de los números primos que aparecen sean pares; con ello se tiene que ese producto es el cuadrado de un número, que denotamos x^2. Por otra parte, el producto de todos los a_i^2, que hacen pareja con los b_i, también es el cuadrado de un número porque lo es cada uno de ellos y este nuevo cuadrado se representa por y^2. Como, por construcción, cada pareja verifica que $a_i^2 \equiv b_i \pmod n$, resulta que se han determinado los cuadrados de dos números, x^2 e y^2, tales que $x^2 \equiv y^2 \pmod n$. Esta es la fase de reducción de la matriz, en que se ha de resolver un sistema lineal de aproximadamente tantas ecuaciones cuantos sean los primos de la base de factores.

Un algoritmo para llevar a cabo esta criba cuadrática, escrito en pseudocódigo, puede encontrarse con más detalle en [MOV97] (§3.2.6). El tiempo de ejecución asintótico, tanto del algoritmo de la criba cuadrática como del de su variante con polinomios múltiples, es $O(\exp(\ln(n)^{1/2}\ln(\ln(n))^{1/2}))$.

EL RSA-130 Y LA CRIBA DEL CUERPO DE NÚMEROS

Conseguida la factorización del número RSA-129, se abordó la factorización del siguiente número (véase http://www.rsa.com/rsalabs/challenges/factoring/ index.html):

RSA-130 = 18070 82088 68740 48059 51656 16440 59055 66278 10251 67694 01349 17012 70214 50056 66254 02440 48387 34112 75908 12303 37178 18879 66563 18201 32148 80557

En este caso se hizo uso de una extensión de la criba cuadrática, llamada la criba del cuerpo de números (NFS, Number Field Sieve, [LLMP93] , [Elk96]). En este método, como en el anterior, se trata de localizar dos números x e y tales que $x \equiv \pm y \pmod n$ y $x^2 \equiv y^2 \pmod n$ pero usando ahora dos bases de factores en lugar de una sola. Una de ellas la forman todos los números primos menores que un valor determinado; la otra está compuesta por los ideales primos de norma menor que alguna cota en el anillo de los enteros de un adecuado cuerpo algebraico de números (en este método se consideran polinomios que no necesariamente tienen que ser cuadráticos).

Existen versiones especiales de este método; por ejemplo, para factorizar números de la forma $n = r^e - s$; para valores pequeños de r y de |s|, se utiliza el método de la criba especial del cuerpo de números (SNFS, Special Number Field Sieve); y para factorizar genéricamente cualquier número entero se emplea el método de la criba general del cuerpo de números (GNFS, General Number Field Sieve).

Utilizando la criba del cuerpo de números y con un gasto computacional de 1.000 MIPS-año (1 MIPS-año es el gasto computacional de una CPU ejecutando 1 Millón de Instrucciones Por Segundo durante todo un año, es decir, $31{,}536 \cdot 10^{12}$ instrucciones de CPU), el 10 abril de 1996 se obtuvieron los dos factores primos del número RSA130:

p = 39685 99945 95974 54290 16112 61628 83786 06757 64491 12810 06483 25551 57243,
q = 45534 49864 67359 72188 40368 68972 74408 86435 63012 63205 06960 09990 44599.

El polinomio empleado en esta factorización fue:

$$57483\ 02248\ 7384\ 05200\ x^5 + 98822\ 61917\ 48228\ 6102\ x^4 - 13392\ 49938\ 91281\ 7668\ 5\ x^3 + 16875\ 25245\ 88776\ 84989\ x^2 + 37599\ 00174\ 85520\ 8738\ x - 46769\ 93055\ 39319\ 05995$$

y su raíz es 12574 41116 84180 05980 468 módulo RSA-130.

El tiempo de ejecución asintótico es $O(\exp(1{,}92 \cdot \ln(n)^{1/3}\ln(\ln(n))^{2/3}))$. Comparando este valor con el dado para la criba cuadrática, se puede afirmar que para números enteros de menos de 350 bits es más rápido el método de la criba cuadrática, debido a su simplicidad; pero para números mayores, es más rápido el de la criba del cuerpo de números.

OTROS MÓDULOS RSA ROTOS: RSA140, RSA155

El siguiente récord conseguido fue la factorización de un módulo de 140 dígitos el 2 de febrero de 1999, también mediante la criba del cuerpo de números:

RSA140 = 21290 24631 82587 57547 49788 20162 71517 49780 67039 63277 21627 82333 83215 38194 99840 56495 91136 65738 53021 91831 67831 07387 99531 72308 89569 23087 34419 36471,

que puede escribirse como el producto de dos primos de 70 dígitos:

p = 33987 17423 02843 85545 30123 62761 38758 35633 98649 59695 97423 49092 93027 71479,
q = 62642 00187 40128 50961 51654 94826 44422 19302 03717 86235 09019 11166 06539 46049.

Los dos polinomios que se utilizaron para determinar la factorización del número RSA140 fueron:

$$f_1(x, y) = 43968\ 20828\ 40\ x^5 + 39031\ 56785\ 38960\ yx^4 - 73873\ 25293\ 89299\ 4572\ y^2x^3 - 19027\ 15324\ 37429\ 88714\ 824\ y^3x^2 - 63441\ 02569\ 44646\ 17913\ 93061\ 3\ y^4x + 31855\ 39170\ 71474\ 35039\ 22235\ 07494\ y^5,$$

$$f_2(x, y) = x - 34435\ 65780\ 92425\ 36951\ 77900\ 7\ y.$$

El primer polinomio se seleccionó debido a que posee dos propiedades: $|f_1(x, y)|$ se mantiene pequeño sobre su región de criba y tiene muchas raíces módulo primos pequeños y módulo potencias de primos. La selección de los polinomios tomó 2.000 horas de CPU sobre 4 SGI Origin 2000 a 250 Mhz. La base de factores para la factorización de este número contenía 1,5 millones de primos. La criba se hizo utilizando 125 SGI Origin 2000 y estaciones Sun a 175 Mhz de media y 60 PC a 300 Mhz de media. La fase de reducción de la matriz, equivalente a la resolución de un sistema de 4,7 millones de ecuaciones, obtenido a partir de la criba, se ejecutó sobre un Cray con 810 Mb de memoria durante 100 horas. En total se necesitaron 2.000 MIPS-año.

El último récord de criptosistema roto ha sido la factorización de un módulo RSA de 155 dígitos (512 bits), producto de dos primos de 78 dígitos:

RSA155 = 10941 73864 15705 27421 80970 73220 40357 61200 37329 45449 20599 09138 42131 47634 99842 88934 78471 79972 57891 26733 24976 25752 89978 18337 97076 53724 40271 46743 53159 33543 33897
= 10263 95928 29741 10577 20541 96573 99167 59007 16567 80803 80668 03341 93352 17907 11307 779
· 10660 34883 80168 45482 09272 20360 01287 86792 07958 57598 92915 22270 60823 71930 62808 643.

El récord fue logrado en agosto de 1999 por un equipo de la Universidad de San Mateo (California). El desafío lo habían lanzado, de nuevo, los Laboratorios RSA. El tiempo necesario para esta factorización ha sido de 5,2 meses, además de otras 9 semanas de cálculos preliminares. El trabajo es el resultado, otra vez, de una computación masivamente paralela, vía Internet, en la que han participado 292 computadores personales (160 estaciones Sun de 175-400 Mhz, 8 procesadores SGI Origin 2000 de 250 Mhz, 120 Pentium II de 300-450 Mhz y 4 CPU Digital/Compaq de 500 Mhz) con un gasto computacional de 8.000 MIPS-año. El algoritmo utilizado fue la criba del cuerpo de números. En la etapa de la criba trabajaron los 292 computadores mencionados; mientras que en la etapa de reducción de la matriz

trabajó el Cray C916 del Centro Académico de Computación de Amsterdam, dotado con 3,2 Gb de memoria, durante 224 horas.

Los dos polinomios empleados en esta factorización se eligieron por tener las mismas propiedades que para el caso RSA140 y fueron:

$$f_1(x, y) = 11937\ 71383\ 20\ x^5 - 80168\ 93728\ 49975\ 82\ yx^4 - 66269\ 85223\ 41185\ 74445\ y^2x^3 + 11816\ 84843\ 00795\ 21880\ 35685\ 2\ y^3x^2 + 74596\ 61580\ 071786\ 44391\ 97430\ 56\ y^4x - 40679\ 84354\ 23621\ 59361\ 91370\ 84050\ 64\ y^5,$$

$$f_2(x, y) = x - 39123\ 07972\ 11680\ 00771\ 31344\ 9081\ y.$$

La selección de los polinomios llevó, aproximadamente 100 MIPS-año, equivalentes a unos 0,4 años de CPU en un SGI Origin 2000 a 250 Mhz.

En http://www.npac.syr.edu/factoring/overview/RSAFCAList.txt puede verse una lista de nuevos y viejos desafíos propuestos por los laboratorios RSA, que incluyen módulos RSA de 10 en 10 dígitos desde un RSA100 hasta un RSA500. Como ejemplo, el último y mayor de estos números propuestos tiene 500 dígitos y es

RSA500 = 18971 94133 74862 66563 30534 74331 72025 27237 18359 19534 28303 18458 11230 62450 45887 07687 60594 32123 47625 76642 74945 54764 41951 54275 86743 20565 93172 54669 94660 49824 19730 16010 38125 21528 54006 88031 51640 16116 23963 12837 06297 93265 93940 50810 77581 69447 86041 72141 10246 41038 04027 87011 09808 66421 48000 25560 45468 76251 37745 39341 82215 49482 12773 35671 73515 34726 56328 44800 11349 40926 44243 84401 98910 90860 32526 78814 78506 01132 07728 71728 19942 44511 32320 19492 22955 42378 98606 63107 48910 74722 42561 73968 03191 69243 81467 62357 12934 29229 99744 11361.

Después de los datos presentados, no es de extrañar que los Laboratorios RSA aconsejen utilizar módulos RSA con un tamaño no inferior a 768 bits o 230 dígitos. Sin embargo, vista la historia, parece que estos datos son relativamente optimistas si se desea una seguridad que vaya más allá de unos pocos años.

EL DISPOSITIVO *TWINKLE*

El dispositivo Twinkle (The Weizmann Institute Key Locating Engine) es la propuesta de Shamir ([Sha99]) para mejorar la factorización de números grandes: supera en varios órdenes de magnitud la velocidad del método de la criba del cuerpo de números. Se trata de un dispositivo optoelectrónico capaz de analizar 100 millones de números enteros grandes y determinar cuáles factorizan completamente sobre una base de factores formada por los 200.000 primeros números primos, todo esto en menos de 10 milisegundos. El coste de este dispositivo se ha establecido en, aproximadamente, el mismo que el de un potente PC o una estación de trabajo, y es entre 500 y 1.000 veces más rápido que el método de la criba cuadrática en la etapa de la criba.

Las ventajas de este dispositivo son claras en lo referente a la fase de la criba, pero eso no supone que la recuperación de la clave sea más sencilla. Analizando cómo sería un ataque contra el módulo RSA140 con Twinkle, habría que tener en cuenta que si cada dispositivo Twinkle es capaz de manejar alrededor de 200.000 primos y un intervalo de criba de alrededor de 100 millones, harían falta 7 dispositivos, según lo dicho anteriormente para el RSA140. Este conjunto de dispositivos sería unas 1.000 veces más rápido que un computador convencional, de modo que la fase de criba llevaría alrededor de 6 días. La matriz requería unos 4 días para ser resuelta, de modo que la reducción pasaría de 33 días a 10. Sin embargo, a pesar de que se reduzca el tiempo requerido en la criba, el problema sigue siendo la reducción de la matriz. En total harían falta $2{,}5 \cdot 10^{18}$ operaciones aritméticas.

Tamaño clave (en bits)	Nº total operaciones (Tiempo total)	Tamaño base factores	Memoria para Criba	Memoria para reducción matriz
428	$5{,}5 \cdot 10^{17}$	600 Kb	24 Mb	128 Mb
465	$2{,}5 \cdot 10^{18}$	1,2 Mb	64 Mb	825 Mb
512	$1{,}7 \cdot 10^{19}$	3 Mb	128 Mb	2 Gb
768	$1{,}1 \cdot 10^{23}$	240 Mb	10 Gb	160 Gb
1024	$1{,}3 \cdot 10^{26}$	7,5 Gb	256 Gb	10 Tb

Tabla comparativa de los recursos necesarios para factorizar módulos de varios tamaños

Se podría analizar ahora cuánto costaría un ataque contra un módulo RSA de 768 bits. Para este número, la fase de criba requeriría 6.000 veces el tiempo empleado para factorizar un módulo de 512 bits y necesitaría de una base de factores 80 veces mayor, lo que incrementaría en la misma proporción el intervalo de criba. Harían falta 1.200 dispositivos Twinkle para este factor base que, además, deberían ser rediseñados para acomodar intervalos de criba mayores, pues Shamir presentó su dispositivo para atacar la factorización de un módulo de 512 bits. La memoria del supercomputador para reducir la matriz se elevaría a 64 Gb y llevaría 24.000 veces más tiempo hacerlo.

En el caso de un módulo de 1024 bits, se necesitaría un tiempo, para la fase de criba, de entre 6 y 7 millones de veces el utilizado con un módulo de 512 bits. El tamaño de la base de factores (y del tamaño de los intervalos) crecería en un factor 2.500. Por tanto, harían falta 45.000 dispositivos para este factor base y 500.000 años para llevar a cabo la criba. La memoria necesaria para manejar la matriz sería de entre 5 y 10 Tb, y harían falta alrededor de 65 millones de veces el tiempo requerido para factorizar el módulo RSA de 512 bits.

Ofrecemos como resumen una tabla comparativa de los recursos necesarios para factorizar módulos de varios tamaños (ver http://www.rsa.com/rsalabs/bulletins /twinkle.html):

CONCLUSIONES

En cuanto al criptosistema DES, hemos puesto en evidencia su actual debilidad frente a modernos enemigos, como el DESCracker y la necesidad de utilizar otros criptosistemas simétricos con espacios de claves más grandes, como son Triple-DES e IDEA.

Por lo que se refiere a RSA, una de sus ventajas sobre DES es que el tamaño de las claves no es fijo, es decir, permite que a medida que pueda comprometerse la factorización de los módulos RSA empleados (incluso con el desarrollo de nuevos dispositivos hardware), se puedan elegir claves de longitudes mayores que mantengan la seguridad del criptosistema. En los años 80, la recomendación habitual era utilizar claves de 512 bits; mientras que hoy se recomienda el uso de claves de 768 bits para usuarios, de 1.024 bits para organismos y empresas y de 2.048 bits para Autoridades de Certificación.

Raúl Durán Díaz
ALCATEL
rduran@alcatel.es
Luis Hernández Encinas
UNIVERSIDAD DE SALAMANCA
encinas@gugu.usal.es
Jaime Muñoz Masqué
jaime@iec.csic.es
Dpto. de Tratamiento de la Información y Codificación
Instituto de Física Aplicada, CSIC

REFERENCIAS

[AGLL95] D. Atkins, M. Graff, A. K. Lenstra and P. C. Leyland, *The Magic Words Are Squeamish Ossifrage*, Proc. Asiacrypt '94, LNCS 917, 263-277, Springer-Verlag, New York, 1995.

[BiSh91] E. Biham, A. Shamir, *Differential cryptanalysis of DES-like cryptosystems*, Journal of Cryptology, 4(1991), 3-72

[EFF98] Electronic Frontier Foundation, *Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design*, O'Reilly & Associates, 1998.

[Elk96] R. M. Elkenbracht-Huizing. Buchmann, J. Loho and J. Zayer, *An Implementation of the Number Field Sieve*, Exp. Math. 5 (1996), 231-252.

[FGHMM97] A. Fúster Sabater, D. de la Guía Martínez, L. Hernández Encinas, F. Montoya Vitini y J. Muñoz Masqué, *Técnicas Criptográficas de Protección de Datos*, RA-MA, Madrid, 1997.

[Kra26] M. Kraitchik, *Théorie des Nombres*, Tome II, Gauthier-Villars, Paris, 1926.

[Lan00] S. Landau, *Standing the Test of Time: The Data Encryption standard*, Notices of the AMS, 47(2000) 341-349.

[LL93] A. K. Lenstra and H. W. Lenstra (eds.),*The Development of the Number Field Sieve* LNM 1554, Springer-Verlag, Berlin, 1993.

[Ley94] P. Leyland, Multiple Polynomial Quadratic Sieve sans Math, disponible en ftp://ftp.ox.ac.uk/pb/math/rsa129/mpqs_sans_math.Z

[LLMP93] A. K. Lenstra, H. W. Lenstra, M. S. Manasse and J. M. Pollard, *The Number Field Sieve*, 11-42, en [LL93].

[Mat93] M. Matsui, *Linear Cryptanalysis Method for DES cipher*, Advances in Cryptology – Eurocrypt '93 Proc., 386-397, Springer-Verlag, 1994.

[MOV97] A. Menezes, P. Van Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Ratón, FL, 1997.

[Pom85] C. Pomerance, *The Quadratic Sieve Factoring Algorithm*, Proc. Eurocrypt '84, LNCS 209, 169-182, Springer-Verlag, 1985.

[Riv77] R. Rivest, *RSA-129-Challenge*, Scientific American, August, 1977.

[RSA78] R. Rivest, A. Shamir and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM 21(1978), 120-126.

[Sch96] B. Schenier, *Applied Cryptography*, John Willey & Sons Inc., New York, 1996.

[Sha99] A. Shamir, *Factoring Large Numbers with the TWINKLE Device* (Extended Abstract), Eurocrypt'99, Rump session.

[Sil87] R. D. Silverman, *The Multiple Polynomial Quadratic Sieve*, Mathematics of Computation 48(1987), 329-340.