Bol. Soc. Esp. Mat. Apl. nº 17 (2001), 13–21.

Números primos y Criptografía

Adolfo Quirós Gracián
Departamento de Matemáticas
Universidad Autónoma de Madrid
28049 Madrid
e-mail: zuazua@eucmax.sim.ucm.es

De vez en cuando aparece en la prensa (*New York Times*, 20-6-90; *El País*, 28-4-94) una noticia anunciando que un grupo de matemáticos ha conseguido factorizar un número entero como producto de primos. ¿ Por qué algo que parece estar al alcance de cualquier escolar merece esta publicidad ?

La respuesta es que en los últimos años los problemas de encontrar números primos y factorizar números enteros han sido aplicados con éxito a la *Criptografía*.

Para proteger la información podemos ocultarla, o transformar el mensaje de forma que sólo el receptor deseado pueda entenderlos. Supongamos que Ana (A) desea transmitir a Beatriz (B) un conjunto $\mathbf{M}$ de mensajes sin que Cristina (C) pueda leerlos. A debe construir otro conjunto $\mathbf{C}$ de *mensajes cifrados* y una función inyectiva f que a cada mensaje m de $\mathbf{M}$ le haga corresponder un mensaje cifrado $c = f(m) \in \mathbf{C}$. En lugar de transmitir m, A transmitirá $f(m)$. La idea es que B conozca la función inversa f^{-1}, de modo que pueda recuperar m a partir de $f(m)$, pero que C ignore f^{-1}.

Es deseable dotar a los conjuntos $\mathbf{M}$ y $\mathbf{C}$ de alguna estructura matemática que haga fácil utilizar funciones. Expliquemos cómo cifraba sus mensajes Julio César, quien utilizaba como conjuntos $\mathbf{M}$ y $\mathbf{C}$ las letras. Hagamos corresponder las 26 letras del alfabeto castellano a los números enteros entre 0 y 25, de manera que $A = 0, B = 1, C = 2, \ldots, Z = 25$. César sustituía la letra m por $f(m) = m + 3$, de modo que $C(= 2)$ era sustituida por $F(= 5 = 2 + 3)$. ¿ Por quién sustituía la Y, dado que $24 + 3 = 27$ es mayor que 25 ? Observemos que el conjunto de los números entre 0 y 25, que llamaremos $\mathbf{Z}/26$, es el conjunto de todos los restos que podemos obtener al dividir por 26, e identifiquemos dos números si dan el mismo resto al dividirlos por 26. Como $27 \equiv 1$ mód. 26 (se lee 27 *es congruente con* 1 *módulo* 26 y es la forma de abreviar que dan el mismo resto al dividir por 26), César sustituía la $Y(= 24)$ por la $B(= 1)$.

Observación 1: Este proceso de "sumar 3" en $\mathbf{Z}/26$ es fácilmente generalizable. Sea n un número entero positivo cualquiera. Llamamos $\mathbf{Z}/n$ a los números entre 0 y $n-1$, y pensamos en ellos como los n restos que se pueden obtener al dividir un número entero entre n. Podemos ahora definir la suma módulo n de dos elementos de $\mathbf{Z}/n$ como el resto resultante al dividir su suma habitual por n (el caso $n = 12$ es la "aritmética del reloj"). Esta suma en $\mathbf{Z}/n$ tiene las mismas propiedades que la suma de números enteros.

Observación 2: Una vez que sabemos cómo cifraba César, es muy fácil descifrar: $f^{-1}(c) \equiv c - 3$ mód. 26. Incluso si César variaba la clave para cifrar, utilizando las distintas funciones f_e definidas por $f_e(m) \equiv m + e$ mód. 26 para cada valor de e en $\mathbf{Z}/26$, el enemigo sólo tenía que averiguar el e utilizado en cada mensaje (lo que en este caso es muy sencillo) para conocer $f_e^{-1}(c) \equiv c - e$ mód. 26.

Con el tiempo, se perfeccionaron los métodos para mantener las comunicaciones a salvo de personas no deseadas, pero para todos ellos valía la observación 2: si uno conoce la clave utilizada para cifrar es "fácil" encontrar la clave para descifrar. Esto plantea al menos dos problemas: i) cada pareja de corresponsales debe tener su propia clave para cifrar, lo que supone que una red de N personas necesite $N(N-1)/2$ claves distintas; ii) si un nuevo miembro desea incorporarse a la red debe previamente acordar claves con cada uno de los N miembros anteriores.

Criptografía de clave pública y firmas digitales

En 1976, Diffie, Hellman y Merkle propusieron un sistema de *Criptografía de clave pública* que resolvía estos dos problemas.

Supongamos que tenemos funciones f_e, dependiendo de una clave para cifrar e, con la propiedad de que, incluso conociendo e, es *imposible en la práctica* encontrar la correspondiente clave para descifrar, d, que nos permite calcular la función inversa. Tales funciones reciben el nombre de *funciones trampa* o bien *funciones de un sólo sentido*. Con ellas, una red de usuarios puede proteger sus comunicaciones como sigue.

La usuaria A elige sus claves para cifrar y descifrar, e_A y d_A, y lo mismo hacen todos los demás. Las claves para cifrar e_A, e_B, e_C, etc. se publican (de ahí el nombre clave pública) en una guía similar a una guía de teléfonos, pero cada usuario mantiene secreta su clave para descifrar. Si la usuaria B quiere enviar un *mensaje* a A no tiene más que mirar en la guía la clave pública de A y enviar $f_A(mensaje)$. AhoraA, ¡ y sólo ella ! conoce la clave para descifrar d_A

y puede recuperar el *mensaje*. Hay que insistir en que C no puede hacer esto porque no se puede encontrar d_A a partir del conocimiento de e_A.

Obsérvese que cada usuario necesita una sola clave y para unirse a la red basta con entrar en contacto con el administrador que edita la guía. Pero surge un problema nuevo: C puede enviar mensajes a A haciéndose pasar por B, ya que toda la información necesaria para hacerlo es pública. Necesitamos encontrar una forma de firmar los mensajes. Podemos inspirarnos en las firmas manuscritas para conseguir *firmas digitales*. Las primeras consisten en el nombre escrito y rubricado de una manera peculiar que sólo el firmante puede reproducir.

La propuesta de Diffie-Hellman-Merkle también incluye algo que únicamente conoce el autor: cómo descifrar. Cuando B escribe a A, debe concluir su mensaje con su firma especial para A, en este caso $f_A(f_B^{-1}(Beatriz))$. Cuando A recibe el mensaje y aplica f_A^{-1} para leerlo obtiene un mensaje comprensible que termina con algo ilegible, $f_B^{-1}(Beatriz)$, la firma de B, de quien se supone que proviene el mensaje. A no tiene más que mirar en la guía para encontrar f_B y comprobar que el mensaje lo ha enviado B, ya que sólo ella podía haber producido una firma que al aplicarle f_B nos diese como resultado *Beatriz*.

El criptosistema RSA

La primera propuesta operativa (y la más empleada) de un criptosistema de clave pública la hicieron en 1978 Rivest, Shamir y Adleman y en su honor se llama *criptosistema RSA*.

Cuando uno aprende a factorizar un número n en el colegio, el método empleado suele ser, esencialmente, ir buscando divisores primos de n. Si uno llega a $\sqrt{n}$ sin encontrar ningún divisor es que n es primo; en otro caso, uno ha encontado un divisor y debe ahora factorizar el cociente. Pero hay métodos mucho más rápidos que éste para factorizar o comprobar la primalidad de un número y, de hecho, estos dos problemas no son equivalentes: es mucho más fácil decidir si un número es primo o compuesto que, sabiendo que es compuesto, factorizarlo.

En el momento actual, es esencialmente imposible factorizar un número de 400 cifras del que se sabe que es producto de dos primos de unas 200 cifras cada uno. Por el contrario, el mayor primo encontrado (el 1 de junio de 1999) tiene 2.098.960 cifras, se conocen más de 5.000 "primos gigantes" (con más de 10.000 cifras), y resulta rutinario encontrar "primos titánicos" (con más de 1.000 cifras). Esta diferencia en la dificultad es lo que utiliza el criptosistema RSA para construir funciones de un sólo sentido.

Para explicar el procedimiento exacto, debemos observar que, igual que hicimos con la suma, podemos definir el producto módulo n de dos elementos

de $\mathbf{Z}/n$ como el resto resultante al dividir su producto habitual por n. Ejemplo: $5 \cdot 8 \equiv 4$ mód. 12. Este producto tiene muchas de las propiedades del producto de números enteros, y en el caso particular de que $n = p$ sea un número primo se comporta exactamente igual que el producto de números racionales, incluyendo la posibilidad de dividir por cualquier elemento de $\mathbf{Z}/p$ distinto del 0. En el caso general, uno puede únicamente dividir por números *primos con* n.

Se tiene además el "Pequeño Teorema de Fermat" (PTF):

Si p es primo y a es un entero cualquiera, entonces $a^p \equiv a$ mód. p.

Euler dio una generalización que, en el caso particular que nos interesa, dice lo siguiente:

Si $n = pq$ es producto de dos primos distintos, a un entero cualquiera, y k un número tal que $k-1$ es divisible por $p-1$ y por $q-1$, entonces $a^k \equiv a$ mód. n.

Estamos ahora en condiciones de explicar cómo funciona el criptosistema RSA. La usuaria A (y todos los demás por su cuenta) busca al azar dos números primos grandes, p_A y q_A, y un número e_A que sea primo con $p_A - 1$ y con $q_A - 1$. Por tanto, A puede encontrar otro número d_A tal que $e_A d_A \equiv 1$ mód. $(p_A - 1)(q_A - 1)$. Por último, A calcula $n_A = p_A q_A$. Todo esto es "fácil" de hacer. Ahora A puede tirar a la basura p_A y q_A y publicar su clave pública para cifrar, el par (n_A, e_A), mientras mantiene secreta su clave para descifrar (n_A, d_A).

Los conjuntos $\mathbf{M}$ y $\mathbf{C}$ de mensajes sin cifrar y cifrados que utiliza A son ambos $\mathbf{Z}/n_A$ (se pueden traducir los mensajes escritos en castellano a este lenguaje sin más que dividirlos en grupos grandes de letras y ver cada grupo como un número de varias cifras escrito en base 26). La función para cifrar es $f_A(m) = m^{e_A}$ mód. n_A mientras la función para descifrar es $f_A^{-1}(c) = c^{d_A}$ mód. n_A. El Teorema de Euler garantiza que estas funciones son inversa una de la otra.

¿ Por qué es este sistema de clave pública ? Porque la única forma de encontrar la clave para descifrar d_A a partir del conocimiento de n_A y e_A es ser capaz de encontrar la factorización $n_A = p_A q_A$, lo que, como ya hemos dicho, es muy difícil.

Primalidad

El PTF permite encontrar primos sin mucho esfuerzo. Dado un número n podemos elegir al azar otro número a entre 2 y $n-1$ y comprobar (es fácil) si a y

n tienen un divisor común. Si es así, hemos encontrado un factor no trivial de n y hemos acabado. En otro caso, si n fuese primo, por el PTF y dividiendo por a deberíamos tener $a^{n-1} \equiv 1$ mód. n. Si esto, que de nuevo es fácil de calcular, es falso, n es necesariamente compuesto. Si se satisface $a^{n-1} \equiv 1$ mód. n, podemos elegir un a distinto y volver a probar. Por desgracia existen algunos números compuestos, llamados números de Carmichael, que actúan como si fuesen primos desde el punto de vista del PTF, por lo que es muy difícil detectarlos de esta manera: habría que encontrar un a que no fuese primo con n, y éstos son escasos.

A mediados de los años 70, Rabin y Miller observaron que si n es primo y $n-1 = s2^r$ con s impar, debemos tener un poco más que el PTF: debe cumplirse que $a^s \equiv \pm 1$ mód. n o que $a^{s2^t} \equiv -1$ mód. n para algún $0 < t < n$. No hay "números de Carmichael" para este "test" de Rabin-Miller y, si n es compuesto, el "test" debe fallar para al menos tres cuartas partes de los posibles a. Así pues, si repetimos el "test" para k valores de a elegidos al azar y n siempre parece ser primo, la probabilidad de que n sea compuesto es menor que $1/4^k$. Podemos así comprobar que n es compuesto o estar *prácticamente seguros* de que n es primo y tenemos un buen *test probabilístico de primalidad*.

Una vez que estamos casi seguros de que n es primo hay métodos (ideados en los años 80) que, con los ordenadores actuales, permiten demostrar en un tiempo razonable la primalidad de números de hasta 1.000 cifras.

Factorización

Llegamos por fin al problema de cómo factorizar un número del que uno de los "tests" anteriores nos ha dicho que es compuesto. A finales de los 60, Brillhart y Morrison recuperan una antigua idea:

Si tuviéramos dos números x e y tales que $x^2 \equiv y^2$ mód. n, es decir, tales que n divida a $x^2 - y^2 = (x+y)(x-y)$, pero con $x \neq \pm y$ mód. n, podríamos asegurar que el máximo común divisor de n y $x-y$ es un factor no trivial de n.

Para encontrar x e y, empezamos por buscar muchos números x_r tales que, poniendo $x_r^2 \equiv t_r$ mód. n, estos t_r se puedan descomponer como producto de primos pequeños. Si tenemos suficientes x_r, podemos asegurar que un producto de algunos de los t_r va a ser un cuadrado que jugará el papel de y^2 y el correspondiente producto de los x_r^2 será x^2. Brillhart y Morrison dieron un método, llamado *de la fracción continua*, que buscaba que los t_r fuesen pequeños y, con él, encontraron la factorización del número de Fermat $F_7 = 2^{2^7} + 1 = 2^{128} + 1$, de 39 cifras.

En 1981, Pomerance sugiere que, para encontrar x_r tales que x_r^2 mód. n factorice como producto de primos pequeños, no hace falta factorizar penosamente los t_r. El proceso que ideó tiene similitudes con la criba de Eratóstenes y se conoce como *criba cuadrática (QS)*. Es el mejor método conocido para factorizar números arbitrarios del tamaño que se puede manejar actualmente en un ordenador.

En 1986, H.W. Lenstra crea un método totalmente nuevo que empleaba herramientas nunca utilizadas anteriormente para factorizar, las *curvas elípticas*, lo que abrió nuevas vías para atacar el problema.

En octubre de 1988, A.K.Lenstra y Manesse utilizan la criba cuadrática para ser los primeros que consiguen factorizar un número de 100 cifras sin utilizar propiedades especiales de éste.

También en 1988, Pollard tiene una nueva idea, que mejora junto a H.W. Lenstra en el año siguiente: queremos tener $x^2 \equiv y^2$ mód. n; pero, ¿ por qué limitarse a trabajar con números enteros ? Trabajando con conjuntos de números más grandes que los racionales, los *cuerpos de números algebraicos*, crean un nuevo método, la *criba en cuerpos de números (NFS)*, especialmente útil para números de la forma $a^b \pm 1$ con a pequeño.

La criba en cuerpos de números se hizo famosa en junio de 1990, cuando A.K. Lenstra y Manasse consiguieron factorizar el número de Fermat $F_9 = 2^{2^9} + 1 = 2^{512} + 1$, de 155 cifras. Esta factorización se consiguió aplicando ingeniosamente una propiedad de los métodos de criba: se pueden buscar factores en varios intervalos simultáneamente. Utilizando la red de correo electrónico reclutaron voluntarios en todo el mundo y encargaron a cada uno de ellos que fuese buscando soluciones en distintos intervalos.

En abril de 1994 un equipo encabezado por A.K. Lenstra consiguió, utilizando la criba cuadrática, con la ayuda de más de 600 voluntarios y tras 8 meses de trabajo que supusieron aproximadamente 5.000 mips-años de cálculo, factorizar RSA-129.

Este número, de 129 cifras como su nombre indica, fue propuesto por Rivest, Shamir y Adleman, lo que explica el resto del nombre, como un reto para los "factorizadores". Lo publicó Martin Gardner en su columna de *Scientific American* en agosto de 1977, y ofreció un premio de 100 dólares a quien encontrase los dos primos en que se descompone, algo que Gardner pensaba que requeriría millones de años. Pero sólo fueron necesarios 17 años, mejoras en los ordenadores y en las comunicaciones y, sobre todo, algunas brillantes ideas matemáticas.

Hay toda una colección de números RSA para factorizar, que constituyen el "Reto RSA", o "RSA Challenge" en su versión original. Cada uno es un

producto de dos primos, buscados para que descomponer el correspondiente número RSA sea especialmente difícil.

Conclusión

Por supuesto uno puede factorizar números cada vez más grandes. Entre los números notables factorizados están RSA-130 (abril de 1996, NFS) y RSA-155, la primera clave RSA de 512 bits (agosto de 1999, NFS); el "record" para la NFS (especial) es un número de 211 cifras (abril de 1999).

Pero esto no pone intrínsicamente en peligro el criptosistema RSA, ya que, mientras los métodos de factorización sigan requiriendo un tiempo sensiblemente superior al que se necesita para buscar números primos de tamaño comparable, siempre se podrán contrarrestar las mejoras en los métodos y en las máquinas, utilizando números primos más grandes. Con los métodos y ordenadores actuales, dos primos de alrededor de 200 cifras cada uno nos dan total seguridad.

El riesgo para RSA es que no se sabe si factorizar requiere realmente un tiempo grande, o simplemente no sabemos hacerlo mejor. Es posible que, en el futuro, una nueva y brillante idea, procedente quizás de las zonas más teóricas de la teoría de números o la geometría algebraica, permita factorizar casi tan fácilmente como se encuentran números primos. Si eso sucediese, el criptosistema RSA pasaría a la historia como una brillante idea que habría quedado obsoleta.

De hecho, muy recientemente ha surgido un método revolucionario de factorización, basado en cambiar radicalmente el tipo de ordenador utilizado. Peter Shor ha demostrado que sería fácil factorizar si uno dispusiese de un *ordenador cuántico*. Pero, de momento, sólo se sabe cómo construir un ordenador cuántico con siete q-bits (que es como se llaman los "bits" cuánticos) mientras que, para factorizar un número grande, sería necesario uno con varios miles de q-bits. El desarrollo de tal aparato supondría la jubilación de RSA, para ser quizá sustituido por la *Criptografía Cuántica*, pero eso es otra historia.

Breve Bibliografía comentada

- Bauer, F.L.: *Decrypted Secrets*, 2nd edition, Springer-Verlag (2000). También cuenta la historia, pero explica muchísimas más Matemáticas que el anterior.
- Cohen, H.: *A Course in Computational Algebraic Number Theory*, Springer-Verlag (1993). Referencia obligada para los métodos computacionales, pero no es para aficionados. Difícil.

- COMAP: *Las Matématicas en la vida cotidiana*, Addison-Wesley/UAM (1999). El capítulo 10 de este fascinante libro trata sobre la Criptografía. Los restantes 21 capítulos no desmerecen.

- Gardner, M.: *Juegos Matemáticos: Claves de nuevo tipo cuyo desciframiento ocuparía unos cuantos millones de años*, Investigación y Ciencia (octubre 1977) 96–101. La versión española del artículo donde se presenta en sociedad RSA, incluido el "Reto".

- Hellman, M.E.: *The Mathematics of public-key Cryptography*, Scientific American, August 1979, 130–139. La criatura presentada por uno de sus progenitores.

- Koblitz, N.: *A course in Number Theory and Criptography*, 2nd edition, Springer-Verlag (1994). Claro y completo, pero requiere una cierta madurez.

- Pastor, J.; Sarasa, M.A.: *Criptografía Digital*, Prensas Universitarias de Zaragoza (1998). Dirigido más bien a ingenieros, pero es muy completo.

- Ribenboim, P.: *The Little Book of Big Primes*, Springer-Verlag (1991). La versión para no especialistas.

- Ribenboim, P.: *The New Book of Prime Number Records*, Springer-Verlag (1995). Todo lo que siempre quiso saber...

- Silverman, J.H: *A friendly Introduction to number theory*, Prentice Hall (1997). Muy interesante introducción a la teoría de números que, en principio, podría entender un alumno de nivel de C.O.U. Contiene todo lo necesario para entender RSA.

- Singh, S.: *Los Códigos Secretos*, Debate (2000). Muy amena historia de la Criptografía, desde los egipcios hasta nuestros días, con breves apuntes matemáticos.

- Shor, P.W.: *Quantum Computing*, Documenta Mathematica, Extra Volume ICM-1998 I (1998), 467-486. El texto ampliado de la conferencia que impartió en el Congreso Internacional de Berlín, donde se le concedió el Premio Nevanlinna. Explica cómo un ordenador cuántico permite factorizar fácilmente y da algunas ideas y referencias sobre Criptografía cuántica. Disponible también en

  `www.mathematik.uni-bielefeld.de/documenta/xvol-icm/ICM.html`

- Stewart, I: *De aquí al infinito*, Crítica (1998). Una joya de la divulgación matemática. Un capítulo trata de primos y Criptografía, y otro de "fácil y dificil". Todo el libro es una delicia.

- Stewart, I: *Juegos Matemáticos: Recibo de compra en Internet*, Investigación y Ciencia (abril 1996) 87–89. Una interesante aplicación de la Criptografía de clave pública.

- Stewart, I: *Juegos Matemáticos: Caza mayor en territorio primo*, Investigación y Ciencia (julio 1997) 87—89. ¿ Qué hay que hacer para encontrar primos grandes ?

- Stewart, I: *Juegos Matemáticos: Cribas en la tierra de los factores*, Investigación y Ciencia (agosto 1997) 88–90. Presenta dos de los más modernos métodos de factorización: la criba cuadrática y la criba en cuerpos de números.

- The prime number page: `http://www.utm.edu/research/primes`. Aquí se puede encontrar información actualizada casi a diario, y conexiones a otras páginas interesantes.

- Internet Prime Net Server: `http://entropia.com/ips`. Para quienes quieran colaborar en la búsqueda de primos enormes.

- La compañía de los inventores de RSA: `http://www.rsa.com`. Información sobre sus productos y sobre el "RSA Challenge".