Las matemáticas del cifrado

Todo el mundo tiene más o menos una idea de lo que es la criptografía. La idea es cifrar un mensaje de alguna manera, de forma que sea ilegible para el que no sepa descifrarlo. Para ello necesitamos dos elementos básicos: un algoritmo de cifrado, y una clave. Pongamos un ejemplo muy sencillo (y clásico, ya que se utilizaba en la antigua Roma). Imaginemos que nuestro algoritmo de cifrado consiste simplemente en transformar una letra en otra, desplazando el alfabeto n posiciones. Es decir, si *n* es 2, tenemos:

A B C D E ... X Y Z C D E F G ... Z A B

Es decir, la «A» pasaría a ser una «C», la «B» una «D», y así sucesivamente. El texto «mensaje» se convertiría en «ñgouclg».El número de posiciones a desplazar sería la clave. En este ejemplo concreto hemos utilizado el 2 como clave, pero podríamos utilizar otro número. Para descifrar el texto, debemos aplicar un algoritmo inverso, con la misma clave. En este ejemplo, el algoritmo inverso sería desplazar las letras en el otro sentido:

A B C D E ... X Y Z Y Z A B C ... V W X

Y así, «ñgouclg» se convertiría en «mensaje». Esto es lo que se conoce como cifrado simétrico, ya que se utiliza la misma clave para cifrar y descifrar. El principal inconveniente de este tipo de cifrados es el distribuir la clave por un canal seguro, de forma que sólo su legítimo destinatario la reciba. Si alguien intercepta la clave, podrá descifrar todos los mensajes.

Existe otro tipo de cifrado, llamado asimétrico, en el que se utilizan dos claves relacionadas, de forma que si se cifra con una, se debe descifrar con la otra, y viceversa. Si mantenemos una de ellas secreta, y sólo entregamos la otra, cualquiera puede cifrar mensajes que sólo yo puedo descifrar. Y al revés, si yo cifro un mensaje, cualquiera puede descifrarlo, pero sabe que sólo yo he podido cifrarlo (no ha sido un impostor). Esto es lo que se conoce como sistemas de clave pública: de la pareja de claves, una se distribuye libremente (clave pública), y la otra se mantiene secreta (clave privada). Estos sistemas son ampliamente utilizados en informática, tanto para cifrar mensajes como para firmalos digitalmente. En efecto, si yo cifro un mensaje con mi clave privada, existe la certeza de que sólo yo he podido cifrar ese mensaje, por lo que es el equivalente a una firma.

¿Cómo se consigue esto? Aquí debemos abandonar los ejemplos sencillos. La criptografía asimétrica se basa en algoritmos no reversibles, es decir, no tienen un algoritmo inverso. Además, tienen la peculiaridad de que una pareja de claves se relaciona de la forma mencionada antes. Si cifro con una, sólo puedo descifrar con la otra. Un punto fundamental es que la relación entre las claves no es evidente, es decir, **no se puede deducir la clave privada a partir de la clave pública**
La base de todo este tinglado son los números primos. Supongo que todo el mundo recuerda lo que es un número primo: un número que sólo es divisible entre 1 y entre sí mismo. Así, 7 es un número primo (sólo es divisible entre 1 y entre 7) y 6 no (es divisible entre 1, 2, 3, y 6). Otra definición importante a tener en cuenta es la de números coprimos Dos números son primos entre sí, o coprimos, si no tienen ningún factor en común salvo el 1, o dicho de otra manera, su máximo común divisor es 1. Así, 6 y 9 no son coprimos, ya que 6 es divisible entre 1, 2, 3 y 6; y 9 es divisible entre 1, 3 y 9. Sin embargo, 8 y 9 sí son coprimos, ya que 8 es divisible entre 1, 2, 4 y 8. Los números 8 y 9 sólo tienen el 1 como factor común. Fijaos también que ni 8 ni 9 son primos, es decir, dos números coprimos, no son necesariamente primos (aunque podrían serlo, y de hecho, podéis deducir que un número primo es coprimo de todos los números menores que él).
Bien, una vez recordadas estas nociones básicas de matemáticas, voy a explicar de forma sencilla uno de los algoritmos de cifrado asimétrico más utilizados: RSA. La generación de la pareja de claves en RSA se hace de la siguiente manera:

1. Buscamos dos números primos distintos (y bastante grandes), a los que llamaremos p y q.
2. Obtenemos el producto de dichos números ($p \cdot q$), al que llamaremos n. Es decir, $n = p \cdot q$.

3. Obtenemos el producto de los dos números primos menos uno, es decir (p-1)(q-1), al que llamaremos z. Es decir, $z = (p-1)(q-1)$. Ésta es la llamada función φ de Euler, y nos indica el número de todos los números coprimos con n, menores o iguales que n.
4. Buscamos un número primo, menor que z, y coprimo con z, es decir, que no sea factor de z, o dicho de otra manera, que z no sea múltiplo de ese número. A este número lo llamaremos e. Tenemos por tanto que z no es divisible por e.
5. Buscamos un número, al que llamaremos d, tal que su producto con e, se pueda dividir entre z, dando como resto 1 (recordáis lo que es el resto de una división ¿no?). O dicho de otra manera, $d \cdot e$ - 1 es divisible entre z.

Una vez hecho esto, resulta que estos números tienen unas propiedades muy interesantes. Si yo cojo un número cualquiera m, y realizo la operación $m^e \ mod \ n$ (donde *mod* se refiere al resto de la división, es decir, calculo el resto de m^e/n), obtengo un número, al que llamaremos c, que cumple lo siguiente: $m=c^d \ mod \ n$. Es decir, si utilizo e, como exponente, obtengo un número, al que si le aplico el mismo algoritmo, pero con d como exponente, me da el número original. Así que ya tenemos nuestra pareja de claves. El par (e, n) sería la clave pública, y el par (d, n) sería la clave privada (recordemos que para un ordenador, toda la información tratable, sea texto, imágenes, audio, vídeo, etc., se reduce a números).
Fijaos que hemos calculado e y d (las claves) a partir de z, pero este último número no lo necesitamos para nada una vez calculadas las claves, y por tanto lo podemos borrar para siempre (al igual que los números p y q). Si conociéramos z, podríamos deducir una clave a partir de la otra, ya que eso es lo que hemos hecho durante la generación de claves (primero generamos e, y luego d a partir de z y e). Y para obtener z, necesitamos factorizar n (es decir, obtener los números primos que lo componen, p y q). Y aquí está todo el meollo de la cuestión. Con nuestro conocimiento actual de matemáticas, tenemos herramientas para saber si un número es primo o no, sin necesidad de factorizarlo. Factorizar un número suficientemente grande, puede llevar siglos aunque utilicemos los ordenadores más rápidos del mundo, mientras que averiguar si un número del mismo tamaño es primo o no, se puede hacer en segundos (o minutos).
Esto quiere decir que si encontramos una forma de factorizar números grandes en poco tiempo, habremos «reventado» el algoritmo RSA. O dicho de otra forma, para reventar el algoritmo, **necesitamos encontrar nuevas técnicas de factorización de números**. Es decir, necesitamos conocimientos de matemáticas que, a día de hoy, nadie tiene. Fijaos que este es un detalle importante. No importa los conocimientos de informática que uno tenga, sino los conocimientos de matemáticas. Otra posible forma de reventar el algoritmo sería desarrollando supercomputadoras millones de veces más rápidas que las actuales, algo que de momento no es posible con la actual tecnología de semiconductores de silicio (tal vez se consiga con futuros ordenadores cuánticos).
¿Cómo de grandes son los números de los que estamos hablando? Pues de cientos de dígitos. Actualmente se utilizan en la mayoría de los casos, números de 1 024 bits, que tienen algo más de 300 dígitos. Pero últimamente se ha puesto en tela de juicio si ese tamaño es suficiente (cada pocos meses aparecen procesadores y ordenadores cada vez más rápidos), por lo que ahora se recomiendan claves de 2 048 bits, lo que nos daría números de más de 600 dígitos. ¿Podéis imaginarlo?

Un ejemplo con números pequeños, para que se entienda bien:

1) Tomamos dos números primos. Por ejemplo $p = 7$ y $q = 13$.
2) El producto es $n = pq = 91$
3) $z = (p-1)(q-1) = 6 \cdot 12 = 72$
4) Primo coprimo con z. Es decir, un numero primo, menor que 72, y que no sea divisor exacto de 72. El primero que encontramos es el 5, ya que 72 : 5 = 14,4, luego $e = 5$.

5) Un numero *d*, tal que $de-1$ sea divisible por z. Buscando, tenemos que si $d=29$, $de=5\cdot 29=145$, que al dividirlo por $z=72$, da resto 2. (Visto de otro modo, $pq-1=144$, que es divisible por z=72).

Ya tenemos nuestra cable publica $(e,n)=(5,91)$ y nuestra clave privada $(d,n)=(29,91)$.

Ahora cifraremos el mensaje $m=32$, por ejemplo.

Para cifrar hacemos

$$m^e \bmod(n) = 32^5 \bmod(91) = 33\ 554\ 432 \bmod(91) = 2 = c$$

que seria nuestro mensaje cifrado.

Para descifrar, recibiríamos *c* y le aplicaríamos

$$m = c^d \bmod(n) = 2^{29} \bmod(91) = 536\ 870\ 912 \bmod(91) = 32$$

con lo que hemos recuperado el mensaje original.

Para romper nuestra clave, tendríamos que ser capaces de factorizar el numero $n=91$, de obtener los dos factores primos que lo componen. 91 es sencillo de factorizar en $7\cdot 13$, pero como se comenta en el texto, para números tan gigantescos como 2 048 bits, la cuestión ya toma otras proporciones, y hay que buscar métodos alternativos.