El sistema RSA

Estímulo del Talento Matemático

Real Academia de Ciencias

30 de julio de 2003

1. Criptografía de clave . . . ¿pública?

Recordemos, de la primera sesión:

- ¿Qué es la criptografía?
- ¿En qué consiste la *criptografía clásica*?
- Problemas que plantea: número de claves e intercambio de las mismas.
- *Intercambio de claves* (Diffie y Hellman) resuelve el segundo problema. Su seguridad se basa en la dificultad de resolver el problema del *logaritmo discreto*.
- ¿En qué consiste la *criptografía de clave pública*?

En los años 70, se produce una revolución en el campo de la criptografía con la aparición de la criptografía de clave pública. En ella, hay dos tipos de claves, ya no sirve la misma para cifrar que para descifrar. (Clásica = simétrica, de clave pública = asimétrica).

Un mensaje para Mortadelo se cifra aplicando un proceso muy sencillo que utiliza su clave pública, y Mortadelo descifra el mensaje deshaciendo ese proceso con una clave privada que sólo él posee.

Es prácticamente imposible deshacer el proceso en cuestión, salvo que se posea la clave privada. Teniéndola es muy fácil.

¿Conoces algún sistema que funcione de esta manera?

Este sistema resuelve el problema del gran número de claves que precisaba la criptografía clásica: un sistema de n usuarios necesita

sólo n claves secretas, una por cada uno de ellos (antes se necesitaban $n\frac{(n-1)}{2}$). Además, el problema del intercambio de claves ya no existe. Las públicas aparecen en una guía y cada usuario recibe su clave privada de las autoridades del sistema.

Hoy en día, la criptografía de clave pública se utiliza en un montón de situaciones reales: controles de acceso (por ejemplo, el sistema de contraseñas para acceder a un ordenador), identificación de personas (por ejemplo, cuando usamos las tarjetas de los cajeros), autentificación (las firmas digitales que garantizan que alguien es quien dice ser electrónicamente).

Que el proceso de cifrado sea "prácticamente imposible" de deshacer quiere decir que el tiempo que llevaría es demasiado largo para que en la práctica permita leer mensajes secretos.

Lo que hay detrás de estos procedimientos son unas funciones matemáticas que suelen llamarse *funciones trampa o ratonera*.

Vamos a experimentar con el criptosistema de clave pública conocido como RSA, cuya seguridad se basa en la dificultad de factorizar números grandes. En pocas palabras, aquí lo **fácil** es **multiplicar** números y lo que es **muy difícil** es el proceso inverso, **encontrar los factores primos** de un número.

2. Lo que necesitamos saber de Maple

Después de iniciar Maple, veremos un símbolo ">"en la pantalla. A continuación de ese símbolo, le iremos dando instrucciones a Maple que terminaremos siempre con un ";"

Las instrucciones que daremos a Maple serán:

- elevar un número a una potencia módulo otro número,

  ```
  > 12111^131 mod 661643;
  ```

  109073

- encontrar el máximo común divisor de dos números,

  ```
  > gcd(131,659880);
  ```

  1

- factorizar en primos un número entero:

  ```
  > ifactor(661643);
  ```

  (541)(1223)

- calcular el inverso módulo un número

  ```
  > 131^(-1) mod 659880;
  ```

  639731

  Recordemos que esto significa que

$$639731 \times 131 \equiv 1 \mod 659880.$$

3. ¿Por qué es difícil factorizar?

Eso de que es muy difícil factorizar un número en primos. . . ¡pero si lo aprendimos hace mucho en el colegio! Mira, por ejemplo, tomo el número 486 y como veo que es par divido por 2 y me queda 243. Este no es par pero enseguida veo que es múltiplo de 3, divido y me queda 81 que me suena mucho porque es $9^2 = 3^4$; y si no me suena da igual, yo sigo dividiendo por 3 y llego a lo mismo, o sea que $486 = 2 \times 3^5$. Ya está, no parece tan difícil hacer esto.

¿Y si el número es 713? No es mucho más grande que el anterior, pero no es par, ni es divisible por 3, ni por 5, ni por 7 (me basta ir probando con los números primos) . . . ni por 11, ni por 13, ni por 17 . . . esto se está poniendo feo . . . ni por 19, ¡ah! menos mal, se puede dividir por 23 y sale 31, que también es primo. En resumen, $713 = 23 \times 31$. Bueno, ha costado algo más, pero lo hemos conseguido.

¿Todavía crees que factorizar es fácil? Pues inténtalo con el número $n = 23360947609$.

¡glub!. . .

El "primer"número primo que divide a este n es $p = 152041$. ¿Cuánto habrías tardado en encontrarlo?

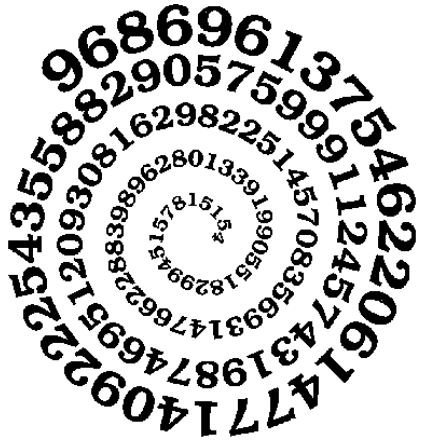
Bueno, pero es cuestión de tiempo, y con los ordenadores actuales el tiempo no es un problema. . .

En realidad, sí lo es y para hacernos una idea, con este método de probar a dividir entre los primos, desde 2 en adelante (observando

que basta con probar hasta llegar a $\sqrt{n}$ aproximadamente), factorizar un número del orden de 10^{12} podría llevarnos 1 segundo, pero si es del orden de 10^{20} ya sería un año y si ponemos 10^{60}, apaga y vámonos, ¡tardaríamos 10^{30} años!

Ya estarás imaginando que debe haber métodos más rápidos para factorizar y es verdad, los hay.

En los años 60, se podían factorizar números de unas 40 cifras sin despeinarse demasiado. A finales de los 80, el récord estaba en unas 100 cifras. A lo largo de los 90, se han factorizado números cada vez más grandes: en 1994 cayó el llamado RSA129 (de 129 cifras) y dos años después, el RSA130 (gracias, entre otras cosas, al trabajo de unos 600 voluntarios que pusieron su granito de arena con sus ordenadores personales).



4. ¿Cómo funciona RSA?

Después de que, en 1976, Diffie y Hellman asombraran al mundo con su sistema de intercambio de claves y propusieran el concepto de cifrado asimétrico, en 1977 aparece el llamado sistema RSA, uno de los primeros criptosistemas de clave pública (y todavía muy usado hoy en día). Su nombre se debe a sus autores Rivest, Shamir y Adleman[1].

Un sistema está formado por unos cuantos usuarios que se comunican entre sí. Pero claro, también hay algún espía que otro que pretende *romper el sistema*, leyendo mensajes ajenos o haciéndose pasar por algún usuario.

El usuario A tiene una **clave pública**: dos números (n, e). El resto de los usuarios utiliza estos dos números para cifrar los mensajes que quiere enviar a A.

[1] Aunque recientemente se ha sabido que el servicio secreto británico había diseñado un sistema semejante unos años antes.

- n va a ser un número (muuuuy grande) producto de dos primos, $n = p \times q$. Atención, los primos p y q sólo los conoce A.
- Una cuenta intermedia: ¿cuántos números más pequeños que un primo p no tienen factores comunes con p? La misma pregunta, para $n = p \times q$. La respuesta es $\phi(n) = (p-1) \times (q-1)$ (recuerda la *función de Euler* $\phi(n)$ de la primera sesión de Criptografía).
- El número e se elige al azar, con la única condición de que no tenga factores comunes con $\phi(n)$ (esto es, el máximo común divisor de e y $\phi(n)$ es 1).

El usuario A, además, dispone de una **clave privada**, un número d que sólo él conoce. Este número es el **inverso** de e módulo $\phi(n)$. Es decir, es el número d que verifica

$$ed \equiv 1 \qquad \text{mod } \phi(n).$$

¿Estamos seguros de que existe ese inverso?

¿Cómo se cifran los mensajes? Queremos mandar a A un mensaje y suponemos que es, simplemente, un número, digamos $m = 12111$.

Así que buscamos en la guía la clave pública de A, los números (n, e). Y realizamos la operación

$$m^e \mod n\,.$$

Digamos que la clave pública es $(661643, 131)$. Entonces, calculamos

```
> 12111^131 mod 661643;
                                    109073
```

El resultado, el número 109073, es lo que se le envía a A (observa que ya no coincide con el número original $m = 12111$).

¿Cómo se descifran los mensajes? El usuario A ha recibido el número 109073 (que, recordemos, es m^e módulo n). Él conoce su clave privada d.

Calculémosla, con ayuda de Maple. Primero, la factorización en primos de $n = 661643$:

```
> ifactor(661643);
                    (541)(1223)
```

Es decir, $661643 = 541 \times 1223$. Recordemos que esto sólo lo sabe A: si hubiéramos elegido un n muy grande, sería imposible hacer esta factorización. Por tanto, $\phi(661643) = 540 \times 1222 = 659880$. Ahora calculamos el inverso de $e = 131$ módulo este número, 659880:

```
>131^(-1) mod 659880;
                    639731
```

La clave privada es, pues, $d = 639731$.

Y calcula:

$$109073^{639731} \quad \text{mod } 661643,$$

que no es otra cosa que

$$(m^e)^d \quad \text{mod } n.$$

El resultado, y parece milagroso, vuelve a ser el mensaje original $m = 12111$ (en realidad no es tan milagroso, claro).

Si queremos que $(m^e)^d \equiv m \mod n$ ¿por qué no se elige d simplemente cómo el inverso de e módulo n?

¿Cómo se firman los mensajes? Somos B y queremos enviar a A un mensaje m, pero además queremos "firmarlo". Ojo, no basta con poner un nombre al final del mensaje, porque eso lo podría hacer cualquiera. Queremos garantizar a A que el mensaje que va a recibir lo hemos mandado, efectivamente, nosotros.

Nuestra firma va a ser un número `firmaB`.

Vamos a suponer que las claves públicas son (n, e_A) y (n, e_B) (el argumento también funcionaría si n_A y n_B fueran diferentes, esto es sólo por simplificar). Las claves privadas las llamaremos d_A y d_B.

Primero mandamos el mensaje m con el procedimiento habitual: enviamos el número m^{e_A} módulo n. Pensemos en que el mensaje m es un largo $email$ en el que, al final, nos despedimos afectuosamente diciendo que somos B.

Y lo acompañamos de una firma "de verdad", una **firma digital**: enviamos el número $(\texttt{firmaB})^{d_B}$ módulo n. Lo podemos hacer porque conocemos d_B, claro.

A recibe m^{e_A} módulo n y lo descifra utilizando su clave privada d_A. Recupera así el $email$ m, en el que se afirma que es B quien lo manda.

Pero A quiere comprobarlo. Así que busca en la guía la clave pública de B, e_B, y calcula

$$\left[(\texttt{firmaB})^{d_B}\right]^{e_B} \qquad \text{mod } n.$$

¿Qué crees que obtiene? ¿Garantiza esto que fue B quien lo envió?

Relación entre RSA y factorización. Por cierto, ¿por qué dijimos que RSA se basa en la dificultad de factorizar? No está claro. Si pudiéramos factorizar, ¿podríamos encontrar la clave secreta de algún usuario?

La clave privada d de un usuario es el inverso de su e módulo $\phi(n)$. Los números n y e son públicos, así que, si a partir de n supiéramos calcular $\phi(n)$, sería muy fácil averiguar la clave privada d a partir de la pública (n, e) y el sistema no valdría para mucho.

Pero resulta que, dado n, conocer $\phi(n)$ "equivale" a conocer la factorización de n en primos, o sea, $n = pq$. Veámoslo.

- Si conocemos la factorización de n, es decir, los primos p y q, entonces obtenemos $\phi(n)$ sin dificultad: es simplemente hacer el producto $(p-1)(q-1)$.
- Y, al revés, si tenemos los números n y $\phi(n)$ podemos calcular p y q. Veamos cómo.
  Conocemos n y $\phi(n)$ y sabemos que

$$\begin{cases} n &= xy \\ \phi(n) &= (x-1)(y-1) \end{cases}$$

  donde x e y son las incógnitas que queremos conocer. Despejamos y de la primera ecuación, $y = n/x$, y la sustituimos en la segunda. Nos queda así que

$$\phi(n) = (x-1)\Big(\frac{n}{x} - 1\Big) \Longrightarrow x\,\phi(n) = (x-1)(n-x)\,,$$

  de donde, reordenando, se llega a $x^2 + x\,\big[\phi(n) - n - 1\big] + n = 0$
  ¿Sabrías obtener x (y luego y)?

5. El juego final

Juego 1. Nos organizamos en grupos de tres, cada uno con un ordenador que disponga de Maple. El equipo A quiere mandar un mensaje (un número) a B. El equipo C está por ahí, espiando.

El grupo B empieza construyendo su clave pública. Elige dos primos grandes, p y q, y calcula $n = pq$. Luego elige e (como explicábamos antes) y calcula su clave privada d. Finalmente, dice en alto la clave (n, e) a todo el que quiera oirlo (incluyendo al espía C). Ya está preparado para recibir mensajes.

El grupo A prepara un mensaje m, que cifra con la clave pública de B, y se lo envía (diciéndolo en alto). C, por supuesto, atento a todo.

B descifra el mensaje sin dificultad. C tiene que apañárselas, mientras tanto, para conocerlo también.

Juego 2. A y B se reunen en secreto para acordar un número $n = pq$. Luego eligen sus respectivos e_A y e_B y de ahí, obtienen sus claves privadas d_A y d_B.

Ahora enviamos las firmas digitales de cada uno de ellos. Digamos que el nombre del grupo A se codifica como 100001 y el de B, como 100002. C, mientras tanto, trata de hacerse pasar por ellos.