Criptología: la llave pública RSA

Imaginate: uno de los clientes de un banco está visitando la página internet de su banco desde su PC en su casa y quiere entrar a su cuenta para saber cuanto dinero tiene. Al picar el boton "ENTRAR A MI CUENTA" le sale una ventanita que le pide su contraseña – un número secreto que le dieron al abrir su cuenta en el banco. Digamos que este número secreto es 17. (En realidad el número suele ser mucho mas grande, involucrando letras tambien, pero la idea no cambia).

Pero es peligroso mandar este número 17 por internet, ya que alguien lo puede interceptar y luego puede entrar a la cuenta del cliente indebidamente. Ni el cliente ni el banco quieren que pase esto. Hay que ocultar, o *codificar*, esta información.

En 1978, basado en teoremas de Fermat y Euler de hace mas de 300 años, Rivest, Shamir y Adleman inventaron el siguiente método de codificación.

El banco le da al cliente (o mas bien a su PC) **las siguientes instrucciones para codificar su mensaje**:

- Te doy dos números, 55 y 7.
- Calcula ahora por favor el resíduo modulo 55 de tu número secreto elevado a la 7ma potencia, y mandame el resultado.

O sea, el cliente multiplica su secreto 17 por su mismo 7 veces, lo que obtiene divide entre 55, y lo que sobra (el resíduo, un número entre 1 y 54), es el mensaje codificado. En este caso es el número 8.

Ahora, ¿cómo recupera el banco el mensaje original del cliente (17) desde el mensaje codificado (8)?

Para esto el banco tiene una "potencia decodificadora": el número 23. El banco calcula el resíduo modulo 55 de 8 elevado a la 23ava potencia, y le sale 17. O sea, multiplicando 8 por su mismo 23 veces y dividir el resultado entre 55, lo que sobra es 17, el mensaje original del cliente.

Ahora el banco puede verificar si el cliente mandó la contraseña correcta y decidir si le deja entrar a su cuenta.

En fórmulas: $M = 17$ (el mensaje secreto del cliente), $N = 55$ y $c = 7$ (los números que el banco manda, publicamente al cliente para codificar su mensaje, el "modulus" y la "llave pública"). $C \equiv M^c \pmod N = 8$ (el mensaje codificado que manda el cliente al banco). El banco recupera el mensaje secreto del cliente al usar $d = 23$ (la potencia decodificadora, o la "llave privada") y calcula $M \equiv C^d \pmod N$.

El punto de todo esto es que estas instrucciones para el cliente, incluyendo los números N y c, son **públicas**; por ejemplo, pueden aparecer en la misma página internet del banco. Sin embargo, la "potencia decodificadora" d (en nuestro caso 23) es secreta, y sin ella es prácticamente imposible recuperar el mensaje original M (el número 17 en nuestro caso).

Para entender esto mejor, tenemos que entender la manera en que el banco diseña los números N, c y d. La receta es la siguiente:

- Escoje dos primos p y q (5 y 11 en nuestro caso, pero en realidad tienen que ser muchos mas grande, con cientos de dígitos).
- Define $N = pq$ (55 en nuestro caso) y $f = (p-1)(q-1)$ (40 en nuestro caso).
- Escoje ahora un c entre 1 y f que no tenga factor común con f (7 en nuestro caso, no tiene factor común con 40).
- Encuentra un d entre 1 y f tal que $cd \equiv 1 \pmod f$ (23 en nuestro caso, ya que $7 \cdot 23 = 161 \equiv 1 \pmod{40}$).
- Los números N y c son públicos. El resto, p, q, f y d es secreto.

El éxito de este método ingenioso de codificación de mensages está basado en que la única manera (que conocemos ahora) de decodificar el mensaje C es encontrar los factores primos p y q de N, para luego poder determinar el f y luego d. Dado N solo, sabemos que en principio, uno podria encontrar sus factores primos p y q, pero resulta que si p y q son muy grandes (digamos de 200 dígitos cada uno), es *practicamente imposible de recuperarlos desde su producto* N (hoy en día).