# Práctica de Aplicaciones de la Aritmética Modular: el RSA

El sistema criptográfico RSA es un método para cifrar y descifrar mensajes de clave pública descubierto en los años 70 y que se sigue usando en multitud de transacciones seguras de todo tipo.

Que sea un sistema de clave pública significa que la clave está dividida en dos partes. La parte necesaria para descrifrar mensajes no se puede calcular a partir de la clave usada para cifrar. Eso permite hacer pública la clave de cifrado para que todos nos puedan escribirnos de forma segura, pero únicamente nosotros (que conocemos la clave de descifrado) seremos capaces de leer los mensajes.

El sistema se basa en la exponenciación modular y en la fórmula de Euler. Para hacer el cifrado y descifrado vamos a suponer que el mensaje es un número $m$ (cualquier mensaje se puede convertir en número o conjunto de números). La clave pública será un número $n$ y la clave privada un número $d$.

Por ejemplo:

In [11]:
```
n = 135750757508665909428509882697558688947562353668799594918
d = 619564659328578668067833300480599438680736667593558730557
m = 1234567  # Poned aquí cualquier número que queráis más pe
```

El cifrado se realizaría utilizando la exponenciación modular
$u = m^{65537} \pmod{n}$

In [12]:
```
u = Zmod(n)(m)**65537
print(u)
```

```
108776337158367560055807473183980561388305036696451942336383
6
```

Ese valor de $u$ los transmitiríamos a través de un mensaje al usuario que debe tener la clave privada $d$ para poder descifrar el mensaje. El descifrado habría que hacerlo calculando $u^d \pmod{n}$

In [13]:
```
u**d
```

Out[13]:
```
1234567
```

Vamos a explicar porqué este método funciona. Tomemos $f = \varphi(n)$ la función $\varphi$ de Euler de $n$.

In [14]:
```
f = 135750757508665909428509882697325550509242215185313291179
```

Por la fórmula de Euler, $m^f = 1 \pmod{n}$.

In [16]:
```
Zmod(n)(m)**f
```

Out[16]: 1

Los valores $d$ y $65537$ están relacionados del siguiente modo:

In [18]:
```
65537*d == 1+29911*f
```

Out[18]: True

Dicho de otra forma, $d$ es el inverso modular de $65537$ módulo $\varphi(n)$. Por eso si calculamos
$(m^{65537})^d = m^{1+29911f} = m \cdot (m^f)^{29911} = m \cdot 1 = m \pmod{n}$.
Esto prueba que el método funciona gracias a la fórmula de Euler.

## Análisis de seguridad del sistema

Hemos dicho que en un sistema de clave pública, conocer la clave pública no permite conocer la parte privada de la clave. En este caso la clave pública es $n$ y la privada es $d$. La relación entre ambas es que $d$ es el inverso modular de 65537 módulo $\varphi(n)$.

El inverso modular se puede calcular fácilmente usando el algoritmo de Euclides, por lo que si conociéramos $\varphi(n)$, podríamos obtener la clave de descifrado. El problema es que $n$ es un producto de dos primos muy grandes, $p$ y $q$ y $\varphi(n) = (p-1)(q-1)$ sólo lo podemos calcular si conocemos los primos $p$ y $q$.

Nosotros no conocemos los primos que forman $n$, sólo conocemos $n$ y para encontrar los primos, tendríamos que factorizar $n$, lo cual es imposible si los dos primos son muy grandes. En este caso hemos tomado unos primos relativamente pequeños (de unas 60 cifras cada uno), pero en realidad se deberían tomar primos de unas 300 cifras cada uno, lo cual hace imposible la factorización.

Únicamente la persona que ha generado la clave $n$, puede conocer los primos y por tanto calcular la clave de descifrado $d$. Eso hace que el sistema sea seguro.

## Un comentario final

Al hacer el cifrado, hemos calculado $m^{65537} \pmod{n}$. ¿Porqué ese exponente en particular? En realidad podríamos tomar cualquier exponente $e$ coprimo con $\varphi(n)$, pero el que se utiliza es éste en particular.

La razón es porque este número tiene una propiedad muy interesante: $65537 = 1 + 2^{16}$. Eso permite calcular $m^{65537} = m \cdot ((((((((((((((((m^2)^2)^2)^2)^2)^2)^2)^2)^2)^2)^2)^2)^2)^2)^2)^2)^2) \pmod{n}$ y usando la exponenciación modular que hemos explicado en clase, cifrar un mensaje queda reducido a 17 multiplicaciones modulares, lo cual es una optimización muy buena.

In [ ]: