

# CIFRADO RSA

## - CIFRADO

- >  $C = M^e \bmod n$

$$n = p q$$

$p, q$  primos

$e$  primo con  $(p-1)(q-1)$

- Vamos a considerar enteros no demasiado grandes para simplificar los cálculos:

```
> p:=nextprime(9999999999999999999000);
p := 9999999999999999999073
> q:=nextprime(9999999999999999999500);
q := 9999999999999999999529
> n:=p*q;
n := 9999999999999999999860200000000000000000436617
> nextprime(999);
1009
```

Elegimos  $e$  primo para facilitar que cumpla la condición " $e$  primo con  $(p-1)(q-1)$ "

```
> e:=1009;
e := 1009
```

Comprobamos que  $e$  es primo con  $(p-1)(q-1)$

```
> gcd(e, (p-1)*(q-1));
1
```

- Sea  $M$  el mensaje a codificar  $M = 48424638452735935832$  ( $M < n$ )  
la codificación será:

```
> M:=48424638452735935832;
M := 48424638452735935832
```

$c(m)$  es la función de cifrado ( $\&\wedge$  es una función que calcula la potencia módulo  $n$ )

```
> c:=m->(m&^e) mod n;
c := m -> `&^(m, e) mod n
```

```
> C:=c(M);
C := 362243465519197692179257861809565126818372201483
```

- El mensaje cifrado será :  $C =$

362243465519197692179257861809565126818372201483



## - SEGURIDAD

- Vamos a factorizar el  $n$  utilizado para el cifrado

```
[ > p:=nextprime(9999999999999999999000);  
                               p := 9999999999999999999073  
[ > q:=nextprime(9999999999999999999500);  
                               q := 9999999999999999999529  
[ > n:=p*q;  
                               n := 9999999999999999999860200000000000000000000436617  
[ > evalf(log10(n));  
                               48.00000002  
tiene 48 dígitos  
[ > ifactor(n);  
                               (9999999999999999999529) (9999999999999999999073)
```

Ha tardado 5185 segundos en un AMD K6 III a 400

```
[ > minutos:=iquo(5185,60,segundos);  
                               minutos := 86  
[ > segundos;  
                               25
```

Lo que quiere decir 1 hora, 26 minutos, 25 segundos

- Si intentamos factorizar un número algo más pequeño el tiempo se reduce drásticamente

```
[ > p1:=nextprime(9999999999999999999000);  
                               p := 9999999999999999999007  
[ > q1:=nextprime(9999999999999999999500);  
                               q := 9999999999999999999557  
[ > n1:=p1*q1;  
                               n := 999999999999999999985640000000000000000439899  
[ > evalf(log10(n1));  
                               38.00000000  
38 dígitos  
[ > ifactor(n1);  
                               (9999999999999999999557) (9999999999999999999007)
```

48 segundos en la misma máquina

```
[ > evalf(5185/48);  
                               108.0208333
```

Al pasar de 38 a 48 dígitos el tiempo de factorización se multiplica por algo más de 108