

89-CRIPT-RSA

March 5, 2018

RSA

El método está descrito en la hoja que se ha repartido en clase, y en el PDF que acompaña esta hoja.

Como queremos encriptar palabras de 5 letras de un alfabeto de 26 necesitamos una clave pública n mayor que el número de tales palabras 26^5 pero menor que 26^6 .

```
In [1]: p = next_prime(26^3+5357)
```

```
In [2]: q = next_prime(26^2+10635);p;q;p*q;26^5;26^6
```

```
Out[2]: 308915776
```

```
In [3]: n = p*q; phi = (p-1)*(q-1)
```

```
In [4]: print n
        print phi
```

```
259578029
```

```
259543776
```

```
In [5]: EX = xgcd(17,phi);EX
```

```
Out[5]: (1, 15267281, -1)
```

```
In [6]: EX[1]*17-phi
```

```
Out[6]: 1
```

La clave pública del usuario es el par $(259578029, 17)$, mientras que su clave privada es 15267281.

```
In [7]: alfb = "ABCDEFGHJKLMNOPQRSTUVWXYZ"
```

```
In [8]: L_alfb = list(alfb);print L_alfb
```

```
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S'
```

```
In [9]: C = "TONT0"
```

```

In [10]: def ord2(c):
          return L_alfb.index(c)

In [11]: def chr2(n):
          return L_alfb[n]

In [12]: COD = [ord2(c) for c in C];COD
Out[12]: [19, 14, 13, 19, 14]

In [13]: m = ZZ(COD,26); m
Out[13]: 6740779

In [14]: ENC = m^17%n;ENC
Out[14]: 246929463

In [15]: DIG = ENC.digits(base=26);DIG
Out[15]: [1, 7, 6, 9, 20, 20]

In [16]: from string import *
          M_ENC = join([chr2(item) for item in DIG],sep="");M_ENC
Out[16]: 'BHGJUJ'

In [17]: M_ENC2 = [ord2(c) for c in M_ENC];M_ENC2
Out[17]: [1, 7, 6, 9, 20, 20]

In [18]: m2 = ZZ(M_ENC2,26); m2
Out[18]: 246929463

```

Como debe ser, $m2$ es el mismo que ENC .

```

In [22]: %time DESENC = m2^EX[1]%n;
          print DESENC

```

```

CPU times: user 2.81 s, sys: 152 ms, total: 2.96 s
Wall time: 2.97 s
6740779

```

```

In [23]: DESENCDIG = DESENC.digits(base=26);DESENCDIG
Out[23]: [19, 14, 13, 19, 14]

In [24]: MENSAJE = join([chr2(item) for item in DESENCDIG],sep="");MENSAJE
Out[24]: 'TONTTO'

In [ ]:

In [ ]:

```