89-CRIPT-RSA

March 4, 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]: EX = xgcd(17,phi);EX
```

```
Out[4]: (1, 15267281, -1)
```

```
In [5]: EX[1]*17-phi
```

```
Out[5]: 1
```

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

```
In [6]: alfb = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

```
In [7]: 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 [8]: C = "TONTO"
```

```
In [9]: def ord2(c):
            return L_alfb.index(c)
```

```
In [10]: def chr2(n):
             return L_alfb[n]
```

```
In [11]: COD = [ord2(c) for c in C];COD

Out[11]: [19, 14, 13, 19, 14]

In [12]: m = ZZ(COD,26); m

Out[12]: 6740779

In [13]: ENC = m^17%n;ENC

Out[13]: 246929463

In [14]: DIG = ENC.digits(base=26);DIG

Out[14]: [1, 7, 6, 9, 20, 20]

In [15]: from string import *
         M_ENC = join([chr2(item) for item in DIG],sep="");M_ENC

Out[15]: 'BHGJUU'

In [16]: M_ENC2 = [ord2(c) for c in M_ENC];M_ENC2

Out[16]: [1, 7, 6, 9, 20, 20]

In [17]: m2 = ZZ(M_ENC2,26); m2

Out[17]: 246929463
```

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

```
In [18]: time DESENC = m2^EX[1]%n;DESENC

CPU times: user 3.12 s, sys: 88 ms, total: 3.2 s
Wall time: 3.2 s


In [19]: DESENCDIG = DESENC.digits(base=26);DESENCDIG

Out[19]: [19, 14, 13, 19, 14]

In [20]: MENSAJE = join([chr2(item) for item in DESENCDIG],sep="");MENSAJE

Out[20]: 'TONTO'
```