RSA

Fabricació de Claus

1. Triem 2 primers p i q molt grans de magnitud semblant, però longitud diferent.
Si tenen una magnitud similar, la "Factorització de Fermat" podria ser eficaç.
Aquesta factorització és eficaç si la diferència $(|p-q|)$ és petita.
Aquesta factorització busca a, b t.q : $a^2 - n = b^2$. Si p, q pròxims és fàcil trobar.
2. Calculem $n = p \cdot q$. Aquest serà el nostre mòdul (serà públic).
3. Es fa servir funció $\lambda(n)$ "Lambda de Carmichael". $\lambda(n) = mcm(p-1, q-1)$
Aquest punt és important. $\lambda(n)$ és el nombre més petit t.q $a^{\lambda(n)} \equiv 1 \pmod p$ i $a^{\lambda(n)} \equiv 1 \pmod q$ on a és coprimer amb p, q. Quant. de més coprimers amb $a, < n$. exponent de xifrat
4. Escollim un e coprimer amb $\lambda(n)$, $1 < e < \lambda(n)$. Aquest és exponent públic.
No ha de ser gaire gran per temes de velocitat, també en reconeixen pocs '1's en binari.
#Normalment es fa servir $2^{16}+1 = 65537_{10} = 0001\ 0000\ 0000\ 0000\ 0001$
El motiu que tingui pocs 1's és pq així es calcula tot més ràpid donat que '0' només fa que desplaçar el nombre i no fa falta sumar (quan es multiplica).
5. Calculem un d que serà l'invers multiplicatiu de e mod $\lambda(n)$. Serà el valor t.q $ed \equiv 1 \pmod{\lambda(n)}$. Aquest d permet que $(m^e)^d \equiv m \pmod n$. Aquest d és privat i permetrà desxifrar els missatges encriptats amb (e, n).

> p, q primers diferents i $ed \equiv 1 \pmod{\lambda(n)} \Rightarrow (m^e)^d \equiv m \pmod n$

Clau Pública: (n, e). Tothom la pot conèixer i serveix per encriptar.
Clau Privada: (n, d). Només emissor de les claus la pot conèixer, serveix desencriptar.
$d, p, q, \lambda(n)$ han de ser secret.

Lambda de Charmichael $\lambda(n)$

Propietats: associativa, ∃ neutre, ∃ invers.

Grup: Estructura algebraica. Conjunt elements + Operació que formen tercer element.
Conjunt de coprimers amb n formen grup multiplicatiu mod n.
$\lambda(n)$ retorna el nombre positiu més petit t.q $\forall$ element del grup complex element$^{\lambda(n)} \equiv 1 \pmod n$.
e ha de ser coprimer amb $\lambda(n)$ pq existeixi l'invers d.

Xifrat

El missatge es codifica numericament (amb UTF-8 per exemple). Si el tamany resultant és més petit que la longitud de n, s'afegeix "padding" per a fer-ho més segur. Si és més gran, es treballa amb blocs de $len(n)$.

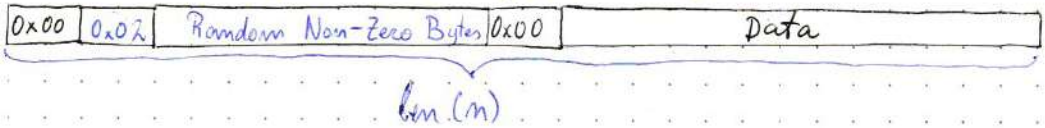
Quan ja tenim $m \Rightarrow c \equiv m^e \pmod n$ on e era exponent públic i $n = p \cdot q$.

Els tamanys de n poden variar: 1024 bits, 2048 bits o fins 4096 bits. # $2048\,b \approx 617$ xifres.

El padding no modifica el missatge i està ben delimitat per a treure-lo.

PKCS#1 v1.5

Estandard comú fet servir per afegir padding en RSA.



⚠ A data 2023 NO es fa servir PKCS #1 v1.5 sinó OAEP. # Però era més difícil d'explicar

Desxifrat

Ara tenim el missatge c que ha estat encriptat amb e. $m^e \equiv c \pmod n$ # és equiv. simplement escrit ad reves

Recordem que $e \cdot d \equiv 1 \pmod{\lambda(n)}$. # d és l'invers i exponit privat.

Per definició significa que $e \cdot d = 1 + \lambda(n)\,k$ on $k \in \mathbb{Z}$

Llavors tenim: $c^d \equiv (M^e)^d \equiv M^{ed} \equiv M^{1+\lambda(n)k} \equiv M \cdot M^{\lambda(n)k} \pmod n$

Recordem també que $a^{\lambda(n)} \equiv 1 \pmod n$. Llavors podem reescriure com:

$c^d \equiv M \cdot M^{\lambda(n)k} \equiv M \cdot (1)^k \equiv M \pmod n$.

Com podem veure ja tenim el que buscàvem, a partir d'un c (missg. xifrat) hem pogut recuperar el M original gràcies a que e, d han sigut generats per la mateixa persona. # En el sentit que no podia ser e, d qualsevol.

Firma

En aquest cas s'eleva a d el hash de M t.q: $H(M)^d \equiv S \pmod n$.

Llavors s'envia M, $H(M)$, S i el receptor fa $S^e \equiv H(M) \pmod n$ i després $H(M)'$. Si $H(M)' = H(M)$ ← Original enviat pel emissor. Sig. que NO a estat alterat.

M que ha rebut → Hash generat pel receptor

Si M modificat, el hash no coincidex.