$\mathbb{Z}_n^*$ elementi invertibili in $\mathbb{Z}_n$

$\mathbb{Z}_n^*$ è un gruppo commutativo

$$\mathbb{Z}_n^* = \{ [a]_n \mid MCD(a,n) = 1 \} \qquad \varphi(n) = |\mathbb{Z}_n^*|$$

$$MCD(a,b)=1 \Rightarrow \mathbb{Z}_{ab}^* \cong \mathbb{Z}_a^* \times \mathbb{Z}_b^*$$

$$(x,y) \longmapsto x\bar{b}b + y\cdot\bar{a}a \quad \text{ove}$$

$\bar{a} = a^{-1} \bmod b$

$\bar{b} = b^{-1} \bmod a$

$$x = \alpha\cdot\beta \bmod ab \iff \begin{cases} x = \alpha\beta \bmod a \\ x = \alpha\beta \bmod b \end{cases}$$

$$MCD(a,b)=1 \Rightarrow \varphi(ab)=\varphi(a)\varphi(b)$$

<u>oss</u> se p primo $\Rightarrow$ $\varphi(p) = p-1$

se $n = p^d$ con p primo $\Rightarrow$ $\varphi(p^d) = p^d - p^{d-1} = p^d\left(1-\frac{1}{p}\right)$

Se $n = p_1^{d_1} p_1^{d_1} \cdots p_k^{d_k}$ con $p_1 \cdots p_k$ primi

$$\Rightarrow \varphi(n) = \prod_{i=1}^{k} \varphi(p_i^{d_i}) =$$

$$= \prod_{i=1}^{k} p_i^{d_i}\left(1-\frac{1}{p_i}\right) = n \prod_{i=1}^{k}\left(1-\frac{1}{p_i}\right)$$

N.B se $r = pq \Rightarrow$ conoscere $\varphi(n)$ implica sapere fattorizzare n

oss: $\forall a \in \mathbb{Z}_n^*$, $[a]^{\varphi(n)} = [1]$ cioè

$$a^{\varphi(n)} \bmod n = 1$$

(in generale un elemento di un gruppo G elevato all'ordine di G fornisce l'identità di G stesso).

RSA: crittosistema asimmetrico a chiave pubblica.

ALICE

BOB

p, q primi

$n = pq$.

$\varphi(n) = (p-1)(q-1)$

fissa e = esponente pubblico

[TROVA d tale che $ed \equiv 1 \bmod \varphi(n)$

↳ alg. euclideo esteso

$ed + k\varphi(n) = 1$

[N.B e deve essere coprimo con $\varphi(n)$]

(e, n) ⟵ (e, n) chiave pubblica

d = chiave privata

m messaggio

$c = m^e \bmod n$ ⟼ c

calcola

$c^d \bmod n = (m^e)^d \bmod n$

$= m^{ed} \cdot 1 \bmod n =$

$= m^{ed} \cdot m^{k\varphi(n)} \bmod n =$

ATTENZIONE! → vale solo se $m \in \mathbb{Z}_n^*$ cioè $MCD(m,n)=1$

[(perché $m^{k\varphi(n)} = 1$)]

$= m^{ed+k\varphi(n)} \bmod n =$

$= m^1 \bmod n = m$

$p, q \approx 10^{100} \Rightarrow n - \varphi(n) = p+q-1 \Rightarrow \boxed{2 \cdot \frac{10^{100}}{10^{200}} = 10^{-100}}$

In realtà la decodifica funziona sempre.

Supponiamo $0 \le x < pq$. Consideriamo

$x^{\varphi(n)} = x^{pq-p-q+1} \bmod n$ $\quad \begin{cases} x^p \bmod p = x \\ x^q \bmod q = x \end{cases}$

$x^{pq-p-q+1} \bmod p =$

$= (x^p)^q \cdot x^{-p} \cdot x^{-q} \cdot x \bmod p =$ $\quad x \in \mathbb{Z}_{pq}^*$ se $x \bmod p \neq 0$, $x \bmod q \neq 0$

$= x^q \cdot x^{-1} \cdot x^{-q} \cdot x = 1 \bmod p$

$x^{pq-p-q+1} \bmod q = 1 \bmod q$

$\Rightarrow x^{\varphi(n)}$ deve soddisfare $\begin{cases} x^{\varphi(n)} = 1 \bmod p \\ x^{\varphi(n)} = 1 \bmod q \end{cases}$

$\Rightarrow$ per CRT $\quad \underline{x^{\varphi(n)} = 1}$

$p \mid x$

$\begin{cases} x = 0 \bmod p \\ x \neq 0 \bmod q \end{cases} \Rightarrow \begin{cases} x^{\varphi(n)} = 0 \bmod p \Rightarrow x^{\varphi(n)} = kp \\ x^{\varphi(n)} = 1 \bmod q \Rightarrow x^{\varphi(n)} \bmod q. \end{cases}$

$\Rightarrow$ per CRT che $x^{\varphi(n)} =$

$\exists!\ 0 \le x^{\varphi(n)} < pq$ con $\begin{cases} p \mid x^{\varphi(n)} \\ x^{\varphi(n)} = 1 \bmod q \end{cases}$ $= pk$ e

$pk \bmod q = 1$

$\Rightarrow k = \bar{p} \bmod q$ cioè k è l'inverso mod q di p.

$x = \bar{p}p \pmod{\varphi(n)}$

posso RICOSTRUIRE IL MESSAGGIO COMUNQUE usando CRT.

perché $p|c$ vale anche $p|m$

$$m^{ed} = m^{ed + k\varphi(n)} = m^{ed} \cdot (\bar{p}p)^k \bmod n$$

$$= \begin{cases} m \bmod q & \text{perché } \bar{p}p = 1 \bmod q \\ 0 \bmod p & \text{perché è multiplo di } p \end{cases}$$

$\Rightarrow$ è proprio m perché il messaggio ritrovato era un multiplo di p (idem se fosse stato un multiplo di q).

DIVISIBILE PER p e congruo a $m \bmod q$

↳ ma m era divisibile per p ed è ovviamente congruo a se stesso mod q] $\Rightarrow$ ho ottenuto proprio m.

In sintesi $c^d = m^{ed} \bmod n = m \bmod n$

$\forall\ 0 \le m < pq$

<u>OSS 1</u>: Bob sceglie $p, q, e \mapsto$ calcola d

$$ed + k\varphi(n) = 1$$

l'esponente pubblico viene di solito fissato e lo si vuole "piccolo"

$$m \longmapsto m^e \bmod n$$

se e piccolo $\Rightarrow d$ grande perché $ed \approx (pq)k$

Il motivo è che la decodifica può essere fatta in modo più

veloce usando CRT; la codifica no → bisogna ricordarsi come lavorare mod n con l'esponente dato!

$$c \longmapsto c^d \bmod n$$

per anni (fino circa al 2006/2007 si è usato $e=3$)

p, q dispari $\quad MCD(e, \varphi(n)) = 1$

$\varphi(n) = (p-1)(q-1)$ pari $\quad$ $e=3$ più piccolo e possibile

è quasi completamente insicuro.

perché è possibile decodificare in modo più veloce

RSA

DEC. STANDARD

$ed + k\varphi(n) = 1$

$c \longmapsto c^d \bmod n$

(lavoro con interi $d, n \approx pq$)

DECODIFICA CON CRT

$ed_1 + k(p-1) = 1 \quad \mathbb{Z}_p^*$

$ed_2 + k(q-1) = 1 \quad \mathbb{Z}_q^*$

$$\begin{cases} m_1 = c^{d_1} \bmod p \\ m_2 = c^{d_2} \bmod q \end{cases}$$

osserviamo che $c^{d_1} \bmod p =$

$= m^{ed_1} \bmod p =$

$= m^{ed_1} \cdot m^{k(p-1)} =$ $\quad$ 0 se $p \mid m$; $m \bmod p$

$= m \bmod p$

Idem per $c^{d_2} \bmod q$.

$$\begin{cases} c^{d_1} = m \bmod p \\ c^{d_2} = m \bmod q \end{cases}$$

CRT $\Rightarrow \exists!\, 0 \le \bar{m} < pq$ con $\bar{m} = m \bmod p$, $\bar{m} = m \bmod q$

ma m soddisfa queste ipotesi $\Rightarrow \bar{m} = m$ ed $\bar{m}$ si ricava da c^{d_2} e c^{d_2}. □

BOB lavora con esponenti $d_1, d_2 \approx p, q$

oss perché tutte funzioni BOB deve conoscere la fattorizzazione di n

→ ALICE NON PUÓ USARE CRT in questo caso.

Il problema di RSA é in effetti DATI (n, e) e c calcolare

$$\sqrt[e]{c} \bmod n$$

$m \longrightarrow c = m^e$ $\qquad$ $c \rightarrow c^{\bar{e}} = m^{e\bar{e}} = m$

N.B il calcolo di d serve proprio ad ottenere queste radici !!

In $\mathbb{Z}_n^*$ le radici e-esime si calcolano trovando d tale che

$$ed + k\varphi(n) = 1$$

(a patto che $MCD(e, \varphi(n)) = 1$).

ATTACCO su e piccolo.

$e = 3$ $\qquad$ p, q primi grandi $\approx 10^{1000}$

$n = pq$

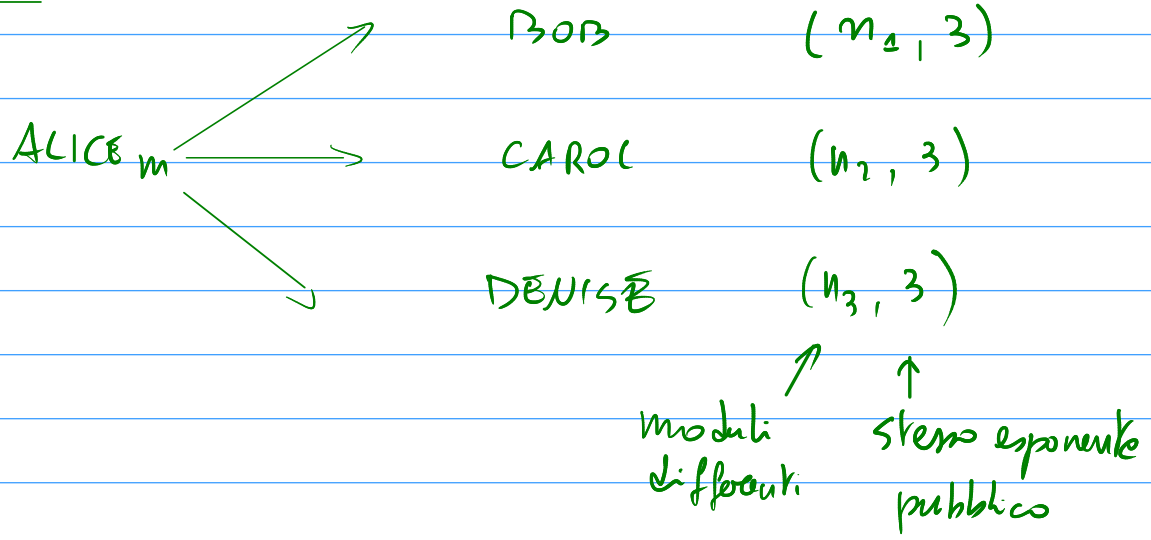
$m^e < n$ può capitare.

$m^e < n \Rightarrow m^e \bmod n = c = m^e$ come intero.

calcoliamo $\sqrt[3]{c}$ negli interi / numeri reali $= m$

In generale se $\sqrt[e]{c}$ è un intero, questo è probabilmente il vostro messaggio !

BROADCAST. $e=3$



$$\begin{cases} c_1 = m^3 \bmod n_1 \\ c_2 = m^3 \bmod n_2 \\ c_3 = m^3 \bmod n_3 \end{cases} \quad \text{si ricava} \quad m^3 > \min(n_1, n_2, n_3)$$

CRT

$$(*) \begin{cases} x = c_1 \bmod n_1 \\ x = c_2 \bmod n_2 \\ x = c_3 \bmod n_3 \end{cases} \rightarrow \begin{cases} \bar{x} = c_1 \bmod n_1 \\ \bar{x} = \bar{c} \bmod n_1 n_2 \end{cases} \rightarrow$$

$$\rightarrow \bar{x} = \bar{c} \bmod n_1 n_2 n_3$$

A) $MCD(n_1, n_2) \neq 1$ o $MCD(n_1, n_3) \neq 1$ o $MCD(n_2, n_3) \neq 1 \Rightarrow$ fine. perché il MCD diverso da 1 ci fornisce un fattore primo di 2 dei moduli $\Rightarrow$

B) $MCD(n_1, n_2) = MCD(n_1, n_3) = MCD(n_2, n_3) = 1$

ricavate i corrispettivi $p, q \rightarrow d$ a partire da e

```
function CRT( a::Vector{T}, b::Vector{T} ) where T <: Integer
    ap,bp,r=my_xgcd(a[2],b[2])
    @assert r≡oneunit(r) "Must be coprime"
    [ mod(a[1]*bp*b[2]+b[1]*ap*a[2],a[2]*b[2]), a[2]*b[2] ]
end

function CRT( x::Vector{Vector{T}} ) where T <: Integer
    reduce(CRT,x)
end
```

$[a, n_1], [b, n_2]$

$\downarrow$

$[a\bar{n}_2 n_2 + b\bar{n}_1 n_1, n_1 n_2]$

OTTENIAMO $\bar{\bar{x}} = \bar{c} \mod n_1 n_2 n_3$ tale che

$$\begin{cases} \bar{\bar{x}} \mod n_1 = c_1 \\ \bar{\bar{x}} \mod n_2 = c_2 \\ \bar{\bar{x}} \mod n_3 = c_3 \end{cases}$$

oss $m^3 > \min(n_1, n_2, n_3)$

D'altro canto $m < \min(n_1, n_2, n_3)$

$\Rightarrow m^3 < \min(n_1, n_2, n_3)^3 \leq n_1 \cdot n_2 \cdot n_3$

osserviamo che $m^3 \mod n_1 = c_1$, $m^3 \mod n_2 = c_2$ $m^3 \mod n_3 = c_3$

$\Rightarrow m^3 = \bar{\bar{x}}$ perché la soluzione del problema del CRT è unica fra $0 \leq \bar{\bar{x}} < n_1 n_2 n_3$

$\Rightarrow m = \sqrt[3]{\bar{\bar{x}}}$ ☐

$e = 65537 = 2^{16} + 1$

RSA gode di una proprietà omomorfica.

ALICE BOB

m, m' messaggi (e, n)

$c = m^e \mod n$

$c' = m'^e \mod n$ $\longrightarrow$ c, c'

EVE può calcolare

$c \cdot c' = m^e \cdot m'^e \mod n =$

$= (m \cdot m')^e \mod n$

DATE 2 CODIFICHE VALIDE → è possibile costruire una codifica

del loro prodotto ed ecco è il prodotto delle codifiche.

N.B poiché RSA è a chiave pubblica, attacchi del tipo chosen plaintext si possono ritenere sempre possibili → in particolare può risultare inutile usare la proprietà omomorfica della codifica

↓

D'altra parte la stessa proprietà vale anche per la decodifica

Sistema in cui abbiamo un ORACOLO che decodifica

∀ messaggio ritorna il messaggio $c = m^e \bmod n$ che ci interessa.

c noto

(e, N) noti.

calcolo $2^e \bmod n = c'$

e poi calcolo

$c'' = c \cdot c' \bmod n$

⟶ $c'' \neq c$

$m' = c''^d \bmod n$

⟵

$m' =$

$= c''^d \bmod n$

$= c^d \cdot c'^d \bmod n = m \cdot 2^{ed} \bmod n$

$= 2m \bmod n$

→ dividiamo per 2 ed otteniamo m.

N.B: DIVIDERE PER 2 vuol dire moltiplicare per x tale che

$2x = 1 \mod n$ → risolvere $2x + kn = 1$ con l'alg. euclideo esteso

Firma digitale con RSA

BOB

$m \longrightarrow s = m^d \bmod n$

ALICE

vede (e, N), m ed s.

e verifica $s^e \bmod n \stackrel{?}{=} m$ — Sì → ACCETTA s; No → RIFIUTA s

$$s^e = (m^d)^e \bmod n = m^{de} \bmod n = m$$

s attesta che — ??

VORREMMO CHE s attestasse la conoscenza di m e di d

mensaggio → (m, s), firma valida di BOB

(m', s')] $\longrightarrow (mm' \bmod n, ss' \bmod n) =$

$= (m'', s'')$

s'' è firma valida su m'' infatti

$s = m^e \bmod n$

$s' = m'^e \bmod n$] $\rightarrow ss' = (mm')^e \bmod n$

N.B La firma con questo algoritmo (cioè "decrittando" il testo in chiaro) si può anche usare come schema di

autenticazione con un crittosistema simmetrico.