ATTACCO VALORI di e

Quando

$$e = \frac{\phi(n)}{2} + 1 \quad \text{oppure} \quad e = \frac{\phi(n)}{k} + 1 \text{ con } k \mid p-1 \text{ e } k \mid q-1$$

allora

$$m^e \bmod n = m \quad (\text{quando } MCD(m,n) = 1)$$

ossia non altera il messaggio

ATTACCO STESSO e

Si suppone di avere e utenti con lo stesso valore di e,

gli e utenti ricevono lo stesso messaggio m

u_1 : $c_1 = m^e \bmod n_1$

u_2 : $c_2 = m^e \bmod n_2$

⋮ ⋮ ⋮

u_e : $c_e = m^e \bmod n_e$

Se non è vero, E trova (p,q).

Hp: $\forall i,j \quad 1 \le i < j \le e \quad MCD(n_i, n_j) = 1$

$\forall i \quad 1 \le i \le e \quad m < n_i$

Sia $n = \prod_{i=1}^{e} n_i$

per il teorema cinese del resto, $\exists!\ m' < n$ t.c.

$$m' = m^e \bmod n,$$

m' si può decifrare in tempo polinomiale

E calcola m'.

$m' = m' \bmod n = m^e \bmod n = m^e$

$m' < n$ ↓ m^e ↓ $= \underbrace{m \cdot \ldots \cdot m}_{e \text{ volte}} < n_1 \cdot \ldots \cdot n_e = n$

Quindi la congruenza diventa

$m' = m^e \Rightarrow m \sqrt[e]{m'}$ Si usa il padding per evitare l'attacco

ATTACCO STESSI n (COMMON MODULUS)

$u_1 : (e_1, n)$

$u_2 : (e_2, n)$

Hp: $MCD(e_1, e_2) = 1$

$u_1 : \quad c_1 = m^{e_1} \bmod n$

$u_2 : \quad c_2 = m^{e_2} \bmod n$

E cerca m in tempo polinomiale

$MCD(e_1, e_2) = 1$

$\Rightarrow \exists r, s$ t.c. $re_1 + se_2 = 1$ Identità di Bezout

Si calcolano con EE in tempo polinomiale

Senza perdita di generalità:

Suppongo $r < 0, \ s > 0$

$m = m^1 = m^{re_1 + se_2} = m^{re_1 + se_2} \bmod n =$

$= (m^{re_1} \bmod n)(m^{se_2} \bmod n) \bmod n =$

$= (m^{e_1} \bmod n)^r (m^{e_2} \bmod n)^s \bmod n =$

$= c_1^r \cdot c_2^s \bmod n$

c_2^s si calcola in tempo polinomiale con l'algoritmo delle quadrature successive

$c_1^r \bmod n = (c_1^{-1})^{-r} \bmod n$

c_1^{-1} è l'inverso moltiplicativo modulo n

A questo punto, si calcola di nuovo con l'algoritmo delle quadrature successive

È necessario che $MCD(c_1, n) = 1$ per risolvere

Se non è vero, allora $c_1 \mid n$, quindi $c_1 = p$ oppure $c_1 = q$

ATTACCHI A TEMPO

Si determina il bersaglio sul tempo impiegato per decifrare; si risolve aggiungendo ritardi casuali.

ESERCIZI SU CIFRARI SIMMETRICI

7) $C_{DES_K}(m, k, w) = w \oplus C_{DES}(m \oplus w, k)$

K chiave DES

w chiave a 64 bit

Come si decifra?

$c = C_{DES_K}(m, k, w) = w \oplus C_{DES}(m \oplus w, k)$

$w \oplus c = C_{DES}(m \oplus w, k)$

$D_{DES}(w \oplus c, k) = D_{DES}(C_{DES}(m \oplus w, k), k)$

$D_{DES}(w \oplus c, k) = m \oplus w$

$m = w \oplus D_{DES}(w \oplus c, k)$

ESERCIZI SU RSA

3) 1. $p = \Theta(\sqrt{n})$, $q = \Theta(n)$

Il cifrario è sicuro? No, sono troppo vicini

2. $p = O(n^{1/3})$, $q = O(n^{2/3})$

È sicuro se $p = \Theta(n^{1/3})$, $q = \Theta(n^{2/3})$

3. $p = \Theta(n^{1/3})$, $q = \Theta(n^{1/3})$

Il problema è minore di n

4. $p = O(\log n)$, $q = O(n \log n)$

p è troppo piccolo

n) Decifrare $c = m^e \bmod n$ in tempo polinomiale

dato $c' = c \cdot x^e$ con $x \in n$ e $MCD(x, n) = 1$.

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

$$= (c^d \cdot x^{ed}) \bmod n =$$

$$= (c^d \bmod n)(x^{ed} \bmod n) \bmod n =$$

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

$$= m \cdot x^{1+k\phi(n)} \bmod n \quad \text{con } k \geq 0$$

$$= m \cdot x \cdot (x^{\phi(n)})^k \bmod n =$$

$$= m \cdot x \bmod n$$

Conosco m', x:

$$m = m' \cdot x^{-1} \bmod n$$

↑ è invertibile per ipotesi: $MCD(x, n) = 1$,

si calcola con EE

m) Decifrare m in tempo polinomiale dati

$k_{pub} = (e, n)$ $\qquad c' = m^{e'} \bmod n$

$k' = (e', n)$, $MCD(e', e) = 1$ $\qquad c = m^e \bmod n$

Si svolge come l'attacco

common modulus.