RSA: sicurezza

Riuscendo a fattorizzare n in tempo polinomiale si potrebbe ricavare la chiave privata da quella pubblica. Non è noto se sono possibili attacchi che non richiedono la fattorizzazione, ma:

- calcolare $m = \sqrt[e]{c} \mod n$ è difficile quanto fattorizzare;
- trovare $\phi(n)$ conoscendo n è equivalente a fattorizzare n;
- ricavare d da (n, e) sembra essere costoso.

p, q e $\phi(n)$ non fanno parte di k_{priv} ma devono rimanere segreti.

La chiave privata d (che dipende da e) deve essere sufficientemente grande da impedire attacchi esaurienti.

Equivalenza tra fattorizzazione e calcolo di $\phi(n)$

- se conosciamo i fattori p e q di n, è facile calcolare $\phi(n) = (p-1)(q-1)$;
- partendo da $\phi(n)$, possiamo trovare in tempo polinomiale p e q:

  $$\phi(n) = (p-1)(q-1) = \overbrace{pq}^{n} - (p+q) + 1,$$

  dunque:

  $$p + q = n - \phi(n) + 1 \qquad (p+q)^2 = (p-q)^2 + 4n$$

  poniamo:

  $$x_1 = p + q \qquad x_2 = p - q,$$

  quindi per quanto trovato prima:

  $$x_1^2 = x_2^2 + 4n \implies x_2 = \sqrt{x_1^2 - 4n},$$

  e:

  $$p = \frac{x_1 + x_2}{2} \qquad q = \frac{x_1 - x_2}{2}.$$

Fattorizzazione

- con General Number Field Sieve (GNFS) il costo è subesponenziale $O(2^{\sqrt{b \log b}})$, quindi non tutti i bit di n sono di sicurezza. Attualmente questo permette di fattorizzare semiprimi fino a circa 768 bit;
- alcuni numeri hanno una struttura particolare che può essere sfruttata per fattorizzare più velocemente, perciò devono essere esclusi nella selezione di p e q;
- fattorizzazione e logaritmo discreto non sono NP-hard, e possono essere risolti in tempo polinomiale su macchine quantistiche.

Bit di sicurezza

A causa dell'algoritmo GNFS, RSA con 2048 bit ha solo 112 bit di sicurezza:

TDEA, AES (bit della chiave)	*RSA* e *DH* (bit del modulo)	*ECC* (bit dell'ordine)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512