una possibile funzione unidirezionale

- moltiplicare due interi a n bit è facile (in $\mathcal{O}(n^2)$ con l'algoritmo usuale)
- trovare un primo a n bit, e verificare che è primo, è facile (vedremo poi)
- fattorizzare un numero a n bit è difficile ($2^{cn^{1/3}}$)
- si può costruire un crittosistema a chiave pubblica basato su questa osservazione?

crittosistema RSA

- Sia $N = pq$, p, q primi. Sia $\mathcal{P} = \mathcal{C} = \mathbb{Z}_N$.
- Lo spazio delle chiavi è

$$\mathcal{K} = \{(N, p, q, d, e) \mid de \equiv 1 \pmod{\phi(N)}\}.$$

- Se $k = (N, p, q, d, e)$ è una chiave, poniamo
- $e_k(x) = x^e \pmod N$
- N e e sono la chiave pubblica
- $d_k(y) = y^d \pmod N$
- p, q, d sono la chiave privata

cifratura e decifratura

$e_k(x) = x^e \pmod N$, $d_k(y) = y^d \pmod N$

- verifichiamo che $d_k(e_k(x)) = x$
- $ed \equiv 1 \pmod{\phi(N)}$, quindi $ed = 1 + k\phi(N)$
- se $(x, N) = 1$, allora

$$\begin{aligned} (x^e)^d &= x^{1+k\phi(N)} \\ &= (x^{\phi(N)})^k x \\ \text{Eulero} \qquad &\equiv 1^k x \pmod N \\ &\equiv x \pmod N \end{aligned}$$

- se $x \notin U(\mathbb{Z}_N)$, vale il corollario già visto (N è il prodotto di due primi)

quindi la funzione di cifratura è iniettiva

implementazione dell'RSA

- Bob genera casualmente due primi "grandi", p e q, con $p \neq q$
- calcola $N = p \cdot q$ e $\phi(N) = (p-1)\cdot(q-1) = pq - (p+q) + 1$
- genera (in maniera non necessariamente casuale) un esponente e con $1 < e < \phi(N)$, tale che $(e, \phi(N)) = 1$
- calcola $d = e^{-1} \pmod{\phi(N)}$
- la chiave pubblica è (N, e), quella privata è (p, q, d)

un esempio

- Bob sceglie $p = 17$ e $q = 11$
- $N = 187$ e $\phi(N) = 16 \cdot 10 = 160$ $(= 2^5 5)$
- Bob sceglie $e = 7$; allora $d = e^{-1} = 23 \pmod{160}$; pubblica la chiave $N = 187$ e $e = 7$
- Alice vuole cifrare il testo in chiaro 88 da mandare a Bob
- calcola $88^7 = 11 \pmod{187}$ e lo invia a Bob
- Bob riceve 11; per decifrare, calcola $11^{23} \pmod{187}$, e ritrova 88

In questo momento, cifratura e decifratura sembrano operazioni computazionalmente impegnative!

Problemi facili

1. dato un intero N, vedere se è primo
2. dati e e M numeri naturali, trovare (e, M); se è 1, calcolare l'inverso di e modulo M (alg. di Euclide – polinomiale)
3. calcolare la f.ne $x \to x^e \pmod{N}$

Problemi difficili

4. dato un intero N, fattorizzarlo
5. dato un intero N, calcolare $\phi(N)$
6. dati N e e, trovare d tale che $(x^e)^d = x \pmod{N}$

problemi facili - calcolare $x \to x^e$

- sia $\mathbf{l}(e)$ la lunghezza di e scritto in base 2 ($\approx \log_2 e$) , sia n la lunghezza di N; $n = \mathbf{l}(N)$; ho $e < N$, quindi $\mathbf{l}(e) \leq n$
- $x^e = x \cdot x \cdot \ldots x$ (e fattori - quindi $\approx 2^{\mathbf{l}(e)}$ fattori - esponenziale)
- l'algoritmo square and multiply calcola $x \to x^e \pmod N$ usando al più $2\mathbf{l}(e)$ moltiplicazioni - e divisioni.
- scrivere e in base 2; $e = 2^{b_1} + 2^{b_2} \ldots 2^{b_k}$, con $b_1 + 1 = \mathbf{l}(e)$
- elevando al quadrato b_1 volte, calcolare $x^2, x^{2^2}, \ldots, x^{2^{b_1}} \pmod N$
- moltiplicando e riducendo $\pmod N$ si ha $x^e = x^{2^{b_1}} \cdot x^{2^{b_2}} \cdot x^{2^{b_k}}$
- con meno di $2\mathbf{l}(e)$ moltiplicazioni – quindi polinomiale

problemi "facili" - primalità

- ci sono infiniti numeri primi (Euclide)
- la dimostrazione è per assurdo (è la più celebre dimostrazione per assurdo): supponiamo ci siano solo un numero finito di primi, e siano questi $p_1, p_2, \ldots, p_k$
- consideriamo il numero $p_1 \cdot p_2 \cdot \cdots \cdot p_k + 1$
- questo numero non è divisibile per nessuno dei $p_1, p_2, \ldots p_k$ – e allora quali sono i suoi fattori primi?
- ci sono infiniti primi – quanti sono i numeri primi che precedono un dato intero x?

teorema dei numeri primi

- $\pi(x)$ = numero di primi p con $p \leq x$
- esempio: $\pi(30) = 4 + 4 + 2 = 10$
- teorema dei numeri primi: asintoticamente, $\pi(x) \approx \frac{x}{\log(x)}$
- congetturato da Legendre, Gauss fine 700 – dimostrato da Hadamard, de la Vallée Poussin fine 800
- dato un intero positivo N, la probabilità che un numero $< N$ scelto a caso sia primo è $\pi(N)/N$, per N molto grande $\approx 1/\log(N)$
- esempio: la probablità che un numero casuale con al più 100 cifre decimali sia primo è $\approx 1/\log(10^{100}) \approx 1/230$
- se sappiamo controllare se un intero è primo, trovare primi grandi è facile

test di primalità

- dato un intero N a n bit, come scoprire se è un primo?
- posso provare a fattorizzarlo per divisioni successive: devo fare al più $\sqrt{N}$ divisioni, quindi è in $\mathcal{O}(2^{n/2})$ – scopro se è primo, e trovo anche i fattori
- i test di primalità scoprono se N è primo senza dire niente dei suoi fattori
- fino al 2002 si conoscevano solo test di primalità polinomiali probabilistici, non deterministici
- nel 2002, M. Agrawal insieme a due suoi dottorandi, Kayal e Saxena, hanno trovato un algoritmo polinomiale per determinare la primalità
- inizialmente, in $\mathcal{O}(n^{12})$ – nel 2005, Lenstra e Pomerance lo portano a $\mathcal{O}(n^6)$
- primes è in P

test deterministici e test probabilistici

- un test deterministico dà una risposta certa: N è primo o non lo è
- un test probabilistico $\mathcal{T}$ consiste in una successione di test $\{\mathcal{T}_m\}_{m\in\mathbb{N}}$ e una successione che va a zero $\{\epsilon_m\}_{m\in\mathbb{N}}$ tale che,
  - se N non passa il test $\mathcal{T}_m$ allora non è primo,
  - la probabilità che N superi i test $\mathcal{T}_1, \ldots, \mathcal{T}_m$ e non sia primo è minore di ϵ_m
- i test di primalità probabilistici più usati sono quello di Solovay-Strassen (1977) e quello di Miller-Rabin (1980)
- il test di Miller-Rabin deriva da un test deterministico (dovuto a Miller) – che però assume l'ipotesi di Riemann generalizzata

idea

- per mostrare che un numero N non è primo si mostra che non si comporta come un primo
- si "cercano prove" del fatto che N non si comporta come un primo
- senza cercare i suoi fattori

PT di Fermat e test di Fermat

- il PTdF dice che, se p è un primo e $1 \leq a \leq p-1$, allora $a^{p-1} \equiv 1 \pmod{p}$
- vogliamo scoprire se N è primo – se trovo $a \leq N-1$ e $a^{N-1} \not\equiv 1 \pmod{N}$ sappiamo per certo che N non è primo – senza sapere niente sui suoi fattori
- questo è il test di Fermat
  - prendo $a < N$, calcolo (a, N); se $\neq 1$, N non è primo
  - se $(a, N) = 1$, calcolo a^{N-1} – se $\not\equiv 1 \pmod{N}$ allora N non è primo
- per esempio, $2^{322} \equiv 157 \pmod{323}$; quindi 323 non è primo $(323 = 17 \cdot 19)$

pseudoprimi

- se però $a^{N-1} \equiv 1 \pmod{N}$ – non posso concludere che N è primo
- per esempio $2^{340} \equiv 1 \pmod{341}$, ma $341 = 11 \cdot 31$
- si dice che N è uno pseudoprimo in base a se N non è primo ma $a^N \equiv a \pmod{N}$. 341 è uno pseudoprimo in base 2.
- possiamo provare diversi valori per a: per esempio $3^{341} \not\equiv 3 \pmod{341}$ – quindi 341 non passa il test!

numeri di Carmichael

- non basta: esistono numeri N che sono pseudoprimi in base a per ogni possibile base a dove $1 < a < N$ e $(a, N) = 1$
- un tale N si dice numero di Carmichael
- per esempio, si può vedere che 561 è un ndC (è il più piccolo)
- il test di Fermat non va bene
- dà comunque un idea di come può funzionare un test di primalità

Problemi facili

1. dato un intero N, vedere se è primo
2. dati e e N, trovare (e, N); se è 1, calcolare l'inverso di e modulo N (alg. di Euclide – polinomiale)
3. calcolare la f.ne $x \to x^e \pmod N$

Problemi difficili

4. dato un intero N, fattorizzarlo
5. dato un intero N, calcolare $\phi(N)$
6. dati N e e, trovare d tale che $(x^e)^d = x \pmod N$

- Se Eve riesce a fattorizzare $N = pq$, ottiene le info private di Bob
- dunque violare l'RSA non può essere più difficile che fattorizzare
- conoscere la fattorizzazione $\iff$ calcolare $\phi(N)$:
  - $\Rightarrow$ ovvio
  - $\Leftarrow$ conoscere N $(= pq)$ e $\phi(N)$ $(= (p-1)(q-1) = N - (p+q) + 1)$ vuol dire conoscere la somma $(N - \phi(N) + 1)$ e il prodotto (N) di p,q – quindi conoscere p e q
- se Eve risolve 6, dati N e e, trova d tale che $(x^e)^d = x \pmod N$
- (Miller) ho buone probabilità di riuscire a fattorizzare N
- quindi – se sospettiamo che Eve ha trovato d, non basta cambiare l'esponente e – bisogna scegliere anche un nuovo N

i problemi 4,5 e 6 sono quindi equivalenti: ma questo non prova che RSA è equivalente alla fattorizzazione – c'è la possibilità di trovare la radice e-sima in qualche altro modo...