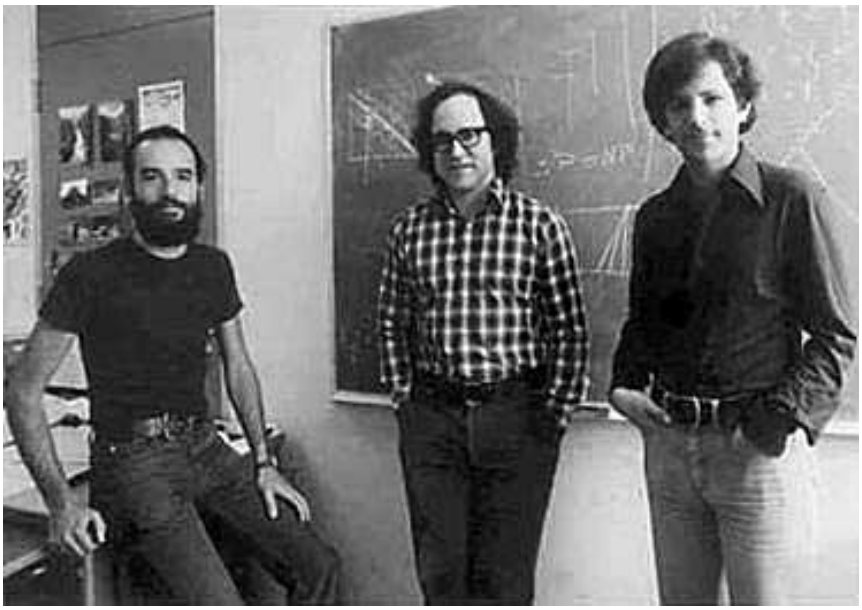
## una possibile funzione unidirezionale

- 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?

## il CS a chiave pubblica RSA - 1978



Shamir, Rivest, Adelman

## 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 $N$, 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 $(x^e)^d = x \pmod N$

Problemi facili

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

## 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