ordine di un gruppo

- G un gruppo finito: ordine di $G = o(G) =$ numero di elementi di G
- l'insieme degli invertibili di $\mathbb{Z}_n$ è un gruppo rispetto al prodotto (mod n)
- si denota con $U(\mathbb{Z}_n)$ e ha ordine $\phi(n)$
- esempio: $U(\mathbb{Z}_9) = \{1, 2, 4, 5, 7, 8\}$, ha $\phi(9) = 6$ elementi

funzione ϕ di Eulero

- funzione ϕ di Eulero, o funzione toziente
- è definita sugli interi positivi
- $\phi(n)$ è il numero di interi positivi $\leq n$ che sono coprimi con n
- $\phi(n) = |\{k \in \mathbb{N}, \quad k \leq n \quad | \quad (k, n) = 1\}|$
  - se p è primo, $\phi(p) = p - 1$
  - se p è primo, $\phi(p^k) = p^k - p^{k-1}$
  - se n, m sono coprimi, allora $\phi(n \cdot m) = \phi(m)\phi(n)$
- in questo modo, si può calcolare la funzione di Eulero di ogni intero
- purché se ne conosca la fattorizzazione
- se $n = p_1^{\alpha_1} \dots p_s^{\alpha_s}$
- $\phi(n) = \phi(p_1^{\alpha_1}) \dots \phi(p_s^{\alpha_s}) = (p_1^{\alpha_1} - p_1^{\alpha_1 - 1}) \dots (p_s^{\alpha_s} - p_s^{\alpha_s - 1})$

ordine di un elemento

- $(G, \cdot)$ un gruppo moltiplicativo di ordine n
- l'ordine di un elemento $g \in G$, $o(g)$, è il minimo intero positivo m tale che
$$g^m = 1$$

- (se $(G, +)$ è un gruppo additivo, l'ordine di un elemento $g \in G$ è il minimo intero positivo m tale che $mg = 0$)
- esempio: in $U(\mathbb{Z}_9)$, calcoliamo $o(2)$
$2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16 = 7, 2^5 = 2 \cdot 7 = 14 = 5,$
$2^6 = 2 \cdot 5 = 10 = 1$ quindi l'ordine di 2 è 6.
- per gli altri elementi, si ha
$o(1) = 1, o(4) = 3, o(5) = 6, o(7) = 3, o(8) = 2$

Teorema (Lagrange)

Se G è un gruppo di ordine n, allora l'ordine di ogni elemento di G divide n

- quindi in particolare si ha $g^n = 1 \quad \forall g \in G$
- se si applica il Teorema di Lagrange al gruppo $U(\mathbb{Z}_n)$ si ottiene

Teorema (Eulero)

Se $(a, n) = 1$, allora $a^{\phi(n)} \equiv 1$ (mod n).

se consideriamo $U(\mathbb{Z}_p)$, p primo, si ha il piccolo teorema di Fermat

Teorema (Fermat)

se p è primo, e $p \not| a$, allora $a^{p-1} \equiv 1$ (mod p)

- quindi se $p \not| a$, si ha $a^p \equiv a$ (mod p)
- anche se $p|a$ si ha $a^p \equiv a (\equiv 0)$ (mod p)
- dunque $\forall a \in \mathbb{N}$ e $m \equiv 1$ mod $p-1$ si ha $a^m \equiv a$ (mod p)

ci servirà il seguente corollario del teorema di Eulero:

Corollario

se $n = pq$, p, q numeri primi, $a \in \mathbb{N}$, sia $m \equiv 1$ (mod $\phi(n)$) allora si ha $a^m \equiv a$ (mod n)

la dimostrazione usa il teorema cinese dei resti

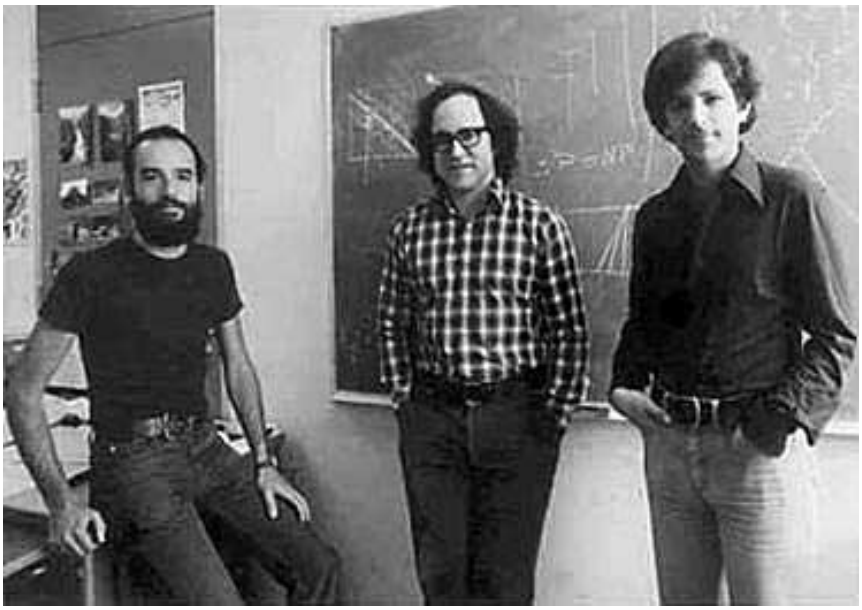
crittosistemi a chiave pubblica

- sviluppare un crittosistema in cui data la funzione di cifratura e_k sia computazionalmente difficile determinare d_k
- Bob rende pubblica la sua funzione di cifratura e_k
- Alice (e chiunque altro) può scrivere a Bob, cifrando il messaggio con la e_k senza bisogno di accordi preliminari
- Bob è l'unico che può decifrare il messaggio
- bisogna che la funzione di cifratura e sia una funzione unidirezionale (one-way function)
- una funzione inveritbile $e : \mathcal{P} \to \mathcal{C}$ si dice unidirezionale se
  - dato $x \in \mathcal{P}$, il calcolo di $e(x)$ è facile
  - per quasi tutti gli $y \in \mathcal{C}$ il calcolo di $e^{-1}(y)$ è difficile

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 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