65206095940827154144915173861033869351028868282615
97291457136781953434590240374747848365524749561797
15310925140079709207334827758905043087098713863654
66626790254909990202656499644185313813356249009490
39987760286127584596268574748390390551023357481383
49987112947978627979534632683398423262500348800198
06468316407450731069455736017920243741207033422885
13860647124387935998503878127495198971160957680515
72029864192570713066673409716032194459919201825563
87385810248187405129722313639926763440877081115833
42725915802423831950700807377335957164110627669686
51596652225167108689682473215490135560006447392 79

=

203956878356401977405765866929034577280193993314348263094772646453283062722701277632936616063144088173312372882677123879538709400158306567338328279154499698366071906766440037074217117805690872792848149112022286332144876183376326512083574821647933992961249917319836219304274280243803104015000563790123

X

319705304701141539155720137200974664667925260594057925396809749294697835128217939956137189431717237652388537524390328359851588290385282149256589183721967420894646839602399199508823558447660553651799376103261276751788573062609555504070444633702398901871897509090368339761978046465893806907794639761 73

Definizione

Un intero $n > 1$ è un numero primo se non esistono due interi $a, b > 1$ tali che $n = ab$.

Sono dunque numeri primi: $2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, \ldots$

Teorema fondamentale dell'aritmetica

Ogni intero $n > 1$ si può scrivere, in modo unico a meno dell'ordine, come prodotto di numeri primi.

Esempio

$140 = 2 \cdot 7 \cdot 5 \cdot 2 = 5 \cdot 2 \cdot 2 \cdot 7 = \cdots$

Di solito si scrive $140 = 2^2 \cdot 5 \cdot 7$.

Teorema

I numeri primi sono infiniti.

Si sa molto di più sulla distribuzione dei numeri primi.
In particolare, il cosiddetto teorema dei numeri primi afferma che i numeri primi fino a n sono (in un senso che si può rendere matematicamente rigoroso) "circa" $\frac{n}{\log(n)}$.

Esempio

I numeri primi fino a un miliardo sono 50847534, mentre

$$\frac{1000000000}{\log(1000000000)} = 48254942,43$$

D'altra parte non si conoscono risultati generali che permettano di determinare facilmente se un dato numero è primo.

Dato un intero $n > 1$ come posso stabilire se n è primo? Nel caso che non lo sia, come posso trovare la sua fattorizzazione?
È facile fornire un algoritmo che risolve entrambi i problemi:

1. $d = 2$
2. se $d > \sqrt{n}$, allora n è primo (fine!)
3. se $d \nmid n$ (d non divide n), sostituisco $d + 1$ a d e torno al punto 2
4. se $d \mid n$ (d divide n), allora n non è primo, d è un suo fattore primo e riapplico l'algoritmo con $\frac{n}{d}$ ($< n$) al posto di n

Osservazione

Nel caso peggiore (cioè quando n è primo), l'algoritmo richiede circa $\sqrt{n}$ passi. Il suo tempo è quindi esponenziale come funzione del numero di cifre di n (che è circa $\log(n)$).

Esistono algoritmi molto più veloci (e complicati...).

- In particolare si può sempre fattorizzare n in tempo subesponenziale, ma non polinomiale (almeno allo stato attuale delle conoscenze).
- È invece possibile stabilire in tempo polinomiale se n è primo o no: anche se questo è stato dimostrato rigorosamente solo nel 2002 da Agrawal, Kayal e Saxena, erano già noti degli algoritmi che in pratica funzionano in tempo polinomiale.
- Inoltre esistono dei test di primalità probabilistici molto semplici e ancora più veloci, che sono sicuri se indicano che n non è primo e altrimenti permettono di concludere solo che n è molto probabilmente primo.

Esponenziali in aritmetica modulare

Per calcolare $a^k \bmod n$ (con $k > 0$) non è necessario calcolare a^k. Se $k > 1$, posso scrivere $k = k_1 + k_2$ on $0 < k_1, k_2 < k$. Essendo

$$a^k = a^{k_1+k_2} = a^{k_1} a^{k_2},$$

se so calcolare $a_i := a^{k_i} \bmod n$, allora

$$a^k \bmod n = (a_1 a_2) \bmod n.$$

Scegliendo $k_1 = k - 1$ e $k_2 = 1$ se k è dispari e $k_1 = k_2 = \frac{k}{2}$ se k è pari, se ne deduce induttivamente che posso calcolare $a^k \bmod n$ con circa $\log(k)$ moltiplicazioni modulari, dunque in tempo polinomiale se $k < n$.

Massimo comun divisore e minimo comune multiplo

Definizione

Dati due interi $a, b > 0$ si definiscono

$$\mathrm{mcd}(a, b) := \max\{k : k \mid a,\ k \mid b\}$$
$$\mathrm{mcm}(a, b) := \min\{k > 0 : a \mid k,\ b \mid k\}$$

Osservazione

- Se sono note le fattorizzazioni di a e di b, si calcolano facilmente $\mathrm{mcd}(a, b)$ e $\mathrm{mcm}(a, b)$. Inoltre

$$\mathrm{mcm}(a, b) = \frac{ab}{\mathrm{mcd}(a, b)}$$

- Si definisce allo stesso modo $\mathrm{mcd}(a, 0)$ e risulta $\mathrm{mcd}(a, 0) = a$ (perché $k \mid 0$ per ogni k).

Algoritmo di Euclide

Se $a \geq b > 0$, per calcolare $\mathrm{mcd}(a, b)$ pongo

$$r_1 := a, \qquad r_2 := b.$$

Poi definisco

$$r_3 := r_1 \mod r_2$$

e induttivamente, se $r_i > 0$,

$$r_{i+1} := r_{i-1} \mod r_i$$

Dato che $r_{i+1} < r_i$, esiste j tale che

$$r_j \neq 0 \qquad \text{e} \qquad r_{j+1} = 0$$

Allora

$$\mathrm{mcd}(a, b) = r_j.$$

Ulteriori proprietà dell'algoritmo Di Euclide

- Poiché $r_{i+1} < r_i < r_{i-1}$, deve essere $q_i > 0$, e quindi

$$r_{i-1} = q_i r_i + r_{i+1} \geq r_i + r_{i+1} > 2r_{i+1}$$

  Ne segue che l'algoritmo di Euclide richiede circa $\log(a)$ passi, e dunque un tempo polinomiale se $a < n$.

- Inoltre, partendo da r_j e sostituendo ricorsivamente r_{i+1} con $r_{i-1} - q_i r_i$, si possono trovare (sempre in tempo polinomiale) due interi x e y tali che

$$ax + by = \mathrm{mcd}(a, b).$$

Esempio

$a = r_1 = 91,\ b = r_2 = 35$

$$
\begin{aligned}
91 &= 2 \cdot 35 + 21 && (q_2 = 2,\ r_3 = 21)\\
35 &= 21 + 14 && (q_3 = 1,\ r_4 = 14)\\
21 &= 14 + 7 && (q_4 = 1,\ r_5 = 7)\\
14 &= 2 \cdot 7 + 0 && (q_5 = 2,\ r_6 = 0)
\end{aligned}
$$

Dunque $j = 5$ e $\mathrm{mcd}(91, 35) = r_5 = 7$. Inoltre

$$
\begin{aligned}
7 &= 21 - 14\\
&= 21 - (35 - 21) = 2 \cdot 21 - 35\\
&= 2 \cdot (91 - 2 \cdot 35) - 35 = 2 \cdot 91 - 5 \cdot 35
\end{aligned}
$$

Teorema cinese del resto

Siano n_1 e n_2 due interi positivi tali che $\mathrm{mcd}(n_1, n_2) = 1$. Dati $0 \leq a_1 < n_1$ e $0 \leq a_2 < n_2$, esiste unico $0 \leq a < n_1 n_2$ tale che

$$\begin{cases} a \mod n_1 = a_1 \\ a \mod n_2 = a_2 \end{cases}$$

Dimostrazione.

Dato che sia a che (a_1, a_2) possono assumere $n_1 n_2$ valori, basta dimostrare l'unicità della soluzione.

Se a e a' sono due soluzioni del sistema, allora $n_1 \mid (a' - a)$ e $n_2 \mid (a' - a)$. Questo implica che

$$\mathrm{mcm}(n_1, n_2) = \frac{n_1 n_2}{\mathrm{mcd}(n_1, n_2)} = n_1 n_2 \mid (a' - a),$$

e quindi $a' = a$ perché $0 \leq a, a' < n_1 n_2$.

Algoritmo RSA

Preliminarmente il destinatario

- sceglie (opportunamente) due numeri primi p e q
- calcola $n := pq$
- calcola $m := \mathrm{mcm}(p-1, q-1)$
- sceglie un intero $1 < c < m$ tale che $\mathrm{mcd}(c, m) = 1$
- trova un intero $0 \leq d < m$ tale che $(cd) \bmod m = 1$
- divulga n e c (la chiave pubblica), mentre tiene segreti p, q, m e d (la chiave privata)

Per mandare un intero $0 \leq a < n$ (ogni messaggio si può trasformare in questa forma, eventualmente spezzandolo) il mittente

- calcola $b := a^c \bmod n$ e manda b

Infine il destinatario

- calcola $b^d \bmod n$ $(= a)$

Sicurezza di RSA

La sicurezza dell'algoritmo dipende da alcune assunzioni su cui non ci sono certezze, ma che sembrano molto plausibili grazie ai numerosi esperti che hanno studiato (e continuano a studiare) la questione. In particolare non deve essere possibile risolvere velocemente nessuno dei seguenti problemi:

- fattorizzare n
- trovare d conoscendo solo n e c
- trovare a conoscendo solo n, c e b

D'altra parte è noto che vanno prese alcune precauzioni tecniche tra cui evitare che a assuma valori particolari per i quali l'ultimo problema si risolve facilmente.

Piccolo teorema di Fermat

Sia p un numero primo e a un intero tale che $a \bmod p \neq 0$. Allora

$$a^{p-1} \bmod p = 1$$

Osservazione

Questo teorema fornisce un criterio molto veloce per vedere se un intero $n > 1$ non è primo: scelto $0 < a < n$, se $a^{n-1} \bmod n \neq 1$, sicuramente n non è primo.
Se invece $a^{n-1} \bmod n = 1$, non si può concludere che n è primo, ma una semplice variante di questo criterio (ripetuto per un numero abbastanza grande di valori di a) permette di sapere che n è molto probabilmente primo.

Perché RSA funziona

p e q primi distinti, $n := pq$, $m := \mathrm{mcm}(p-1, q-1)$, $c, d > 0$ tali che $(cd) \bmod m = 1$. Devo dimostrare che dato $0 \leq a < n$ e posto $b := a^c \bmod n$, vale $b^d \bmod n = a$.
Essendo $b^d \bmod n = (a^c)^d \bmod n = a^{cd} \bmod n$, devo quindi dimostrare $a^{cd} \bmod n = a$, che per il teorema cinese equivale a

$$\begin{cases} a^{cd} \bmod p = a \bmod p \\ a^{cd} \bmod q = a \bmod q \end{cases}$$

Per simmetria basta dimostrare la prima uguaglianza.
Se $a \bmod p = 0$, anche $a^{cd} \bmod p = 0$.
Se $a \bmod p \neq 0$, osservo che $cd = 1 + (p-1)k$ per qualche k (perché $(cd) \bmod m = 1$ e $(p-1) \mid m$), e quindi $a^{cd} = a(a^{p-1})^k$.
Usando il piccolo teorema di Fermat concludo che

$$a^{cd} \bmod p = (a(a^{p-1})^k) \bmod p = (a \cdot 1^k) \bmod p = a \bmod p$$

DIMOSTRAZIONE RSA

TEOREMA CINESE DEL RESTO

n_1, n_2 tali che $\text{mcd}(n_1, n_2) = 1$
a_1, a_2 tali che $0 \le a_1, a_2 < n_{1,2}$
$\Rightarrow \exists!\ a$ tale che $\begin{cases} a \bmod n_1 = a_1 \\ a \bmod n_2 = a_2 \end{cases}$

PICCOLO TEOREMA DI FERMAT

p primo
a tale che $a \bmod p \neq 0$
$\Rightarrow a^{p-1} \bmod p = 1$

RSA

p, q primi distinti
$n = p \cdot q$
$m = \text{mcm}(p-1, q-1)$ (B)
c, d tale che $\text{mcd}(m, c) = 1$ e $cd \bmod m = 1$ (A)
dato $0 \le a < n$ e $b = a^c \bmod n$
$\Rightarrow b^d \bmod n = a$

DIM:

$$b^d \bmod n = (a^c \bmod n)^d \bmod n = a^{cd} \bmod n$$

$$a^{cd} \bmod n = a \iff \begin{cases} a^{cd} \bmod p = a \bmod p & (1) \\ a^{cd} \bmod q = a \bmod q & (2) \end{cases}$$

per teorema cinese del resto e perché $a^{cd} = a + hm = a \cdot k \cdot p \cdot q$

(1) se $a \bmod p = 0 \Rightarrow a^{cd} \bmod p = 0 \Rightarrow 0_p$
se $a \bmod p \neq 0$ $cd = 1 + (p-1)k$ da (A) e (B) $\Rightarrow a^{cd} = a \cdot (a^{p-1})^k$

$$a^{cd} \bmod n = (a \bmod p) \cdot (a^{p-1} \bmod n)^k = a \bmod p \ (1)^k = a \bmod p$$ per il piccolo teorema di F.

(2) analogo a (1)

Esercizio RSA

1. Scegli p tra 2, 3, 5 e q tra: 7, 11, 13
2. Calcola n = p q
3. Calcola m = mcm(p−1,q−1)
4. Scegli un intero c : 1 < c < m tale che
   mcd(c,m) = 1
5. Trova un intero d: 0 ≤ d < m tale che
   (cd) mod m=1
   Per trovare d devi risolvere l'equazione:
   c d – k m = 1 tramite l'algoritmo di Euclide.
6. Divulga n e c (la chiave pubblica), mentre tieni segreti p, q, m e d (la chiave privata)
7. Per mandare un intero 0 ≤ a < n (ogni messaggio si può trasformare in questa forma, eventualmente frammentadolo) il mittente calcola b = (a^c) mod n e manda b
8. Infine il destinatario calcola (b^d) mod n (= a) .

p =		q =	
n =			
p - 1 =		q - 1 =	
m =			
c =			
d =			
CHIAVE PUBBLICA			
n =		c =	
CHIAVE PRIVATA			
m =		d =	