153. Dai Numeri Primi alla Crittografia

Gianluca Salvalaggio
gianluca.salvalaggio@gmail.com

Per secoli il fascino dei numeri primi ha catturato l'interesse dei più grandi matematici. Le proprietà di questi eleganti oggetti sono state a lungo esplorate e in questi ultimi decenni, con l'evoluzione delle tecnologie informatiche, hanno trovato estese applicazioni nell'ambito della cosiddetta Crittografia a chiave pubblica.
Nelle pagine che seguono cercheremo di capire quali profonde problematiche si celano dietro la bellezza dei numeri primi e comprenderemo come essi vengono applicati nelle moderne tecniche crittografiche. In particolare verrà analizzato l'algoritmo di cifratura RSA che sfrutta, con ingegnosa semplicità, la difficoltà di scomporre in fattori primi un numero N molto grande.

1. Numeri primi

I concetti di numero naturale e di numero intero ci sono ben familiari. I numeri naturali sono quelli che si impararano da bambini e che quotidianamente utilizziamo per *contare le cose* ("*… 44 gatti in fila per 6 col resto di 2 …*"). I numeri interi invece vengono informalmente definiti come l'unione dei numeri naturali e dei "numeri con segno" (detti anche numeri relativi). Se quindi indichiamo con $\mathbf{N}$ e $\mathbf{Z}$ rispettivamente l'insieme dei numeri naturali e dei numeri interi, possiamo scrivere:

$$\mathbf{N} = \{ 0, 1, 2, 3, \dots \}$$

$$\mathbf{Z} = \{ \dots, -3, -2, -1, 0, 1, 2, 3, \dots \}$$

Una peculiarità che accomuna i numeri naturali ed i numeri interi è il concetto di divisibilità. Dati i numeri interi m e n, si dice che ***n* divide *m***, e si scrive $n \mid m$, se esiste un intero k tale che $m = nk$. Sia n che k vengono detti divisori (o fattori) di m.
Altresì, dati due numeri m_1 e m_2, si definisce **massimo comun divisore** (*greatest common divisor*), e si indica con $\gcd(m_1, m_2)$, il più grande intero che divide sia m_1 che m_2.
Ad esempio $\gcd(6,8) = 2$, mentre $\gcd(12,4) = 4$.
Siamo in grado, quindi, di definire cos'è un **numero primo**:

"un numero intero m si dice primo se è divisibile solo per se stesso e per 1"

Più semplice di così: i numeri primi non ammettono divisori diversi da loro stessi e da 1.
Alcuni esempi sono: 2, 3, 5, 7, 11, 13, 17, 19, … 53, ..., 101, ….
Viceversa, i numeri interi che non sono primi vengono detti **compositi**, proprio perché possono essere espressi come prodotto di due o più numeri. Ad esempio 6 è esprimibile come $2 \cdot 3$.
Una proprietà un po' più *debole*, e che non riguarda un singolo numero ma una coppia di numeri è la **coprimalità**:

"due numeri m ed n sono coprimi, o primi relativi, se vale $\gcd(m,n) = 1$"

Ad esempio i numeri 8 e 9 presi singolarmente sono tutt'altro che primi ma, come si può facilmente verificare, sono coprimi perché vale $\gcd(8,9) = 1$.

2. Conoscere i numeri primi

Bene, introdotte le doverose definizioni possiamo affrontare alcuni quesiti che riguardano i numeri primi.

- anzitutto, perché sono così importanti?
- sono finiti o infiniti?
- siamo in grado di contarli?

Andiamo con ordine: chiariamo perché è importante studiare i numeri primi.
Si può dire che i numeri primi rappresentano i "mattoni elementari", gli atomi, con cui costruire, attraverso l'operazione di moltiplicazione, tutti gli altri numeri interi. Tale concetto è ben formalizzato nel noto:

Teorema Fondamentale dell'Aritmetica. Ogni intero positivo N o è primo oppure è esprimibile come prodotto di numeri primi, e tale fattorizzazione è unica.
Cosa significa? Significa che preso un qualunque numero naturale, ad esempio 12, esso può essere espresso come prodotto di due o più numeri primi, e tale rappresentazione è unica. Infatti 12 è rappresentabile come il prodotto: $2 \cdot 2 \cdot 3$. Ovviamente se il numero considerato è già primo, ad esempio 7, la sua fattorizzazione è banale.

Al secondo quesito (*sono finiti o infiniti?*) rispose per primo Euclide, dimostrando il seguente:

Teorema sull'Infinità dei numeri primi (Euclide). Esistono infiniti numeri primi.
Dimostrazione. Supponiamo per assurdo che i numeri primi siano in numero finito, diciamo k:
$p_1, p_2, p_3, \ldots, p_i, \ldots p_k$. Consideriamo quindi i numeri $P = p_1 p_2 \cdots p_k$ ed $N = P + 1$. Siccome vale $N > p_k$, il numero N non può essere primo (sarebbe in contraddizione con le ipotesi), quindi deve ammettere dei fattori primi. D'altra parte N non risulta divisibile per nessuno dei primi p_i considerati. Se così fosse, infatti, si avrebbe che p_i divide sia P (per costruzione) che $N = P + 1$, e quindi dividerebbe anche la loro differenza cioè $(P + 1) – P = 1$. Assurdo

Ricapitolando quindi, i numeri primi sono infiniti e per mezzo di loro possiamo esprimere tutti i numeri naturali. A questo punto è lecito chiedersi se siamo in grado di calcolarli o quanto meno di determinare quanti sono i numeri primi minori di un certo valore n.
Queste problematiche sono state affrontate già ai tempi degli antichi Greci, come testimonia il seguente procedimento, attribuito ad Eratostene (III° secolo a.C.)

Crivello di Eratostene. Si tratta di un metodo veloce per determinare i numeri primi minori di una certa quantità n.
L'algoritmo parte dal presupposto che, dato un numero n, il suo più piccolo fattore primo è sempre minore o uguale a $\sqrt{n}$. La cosa è facilmente dimostrabile: sia p il più piccolo fattore primo di n, quindi si può scrivere $n = pd$, con $d \geq p$. Pertanto $n = pd \geq pp = p^2 \Rightarrow p \leq \sqrt{n}$.
Ad esempio, dato $n = 35 = 5 \cdot 7$ si verifica facilmente che 5 è minore di $\sqrt{35}$.
Il Crivello calcola i numeri primi minori di n nel modo seguente:

a. elenchiamo tutti i numeri compresi fra 2 ed n
b. eliminiamo tutti i multipli di 2
c. il più piccolo numero maggiore di 2 "sopravissuto" all'eliminazione precedente è 3: quindi eliminiamo tutti i multipli di 3.
d. si continua così, eliminando i multipli di 5 e poi di 7 etc. etc., fino ad arrivare a $\sqrt{n}$
e. i numeri "sopravissuti" alla selezione (2, 3, 5, …) sono i primi $\leq n$.

3. Il Teorema dei Numeri Primi

La tecnica di Eratostene è semplice, concettualmente chiara, ma impraticabile per valori di n molto grandi. Nei secoli a venire, peraltro, gli sforzi dei matematici si sono concentrati nel cercare di capire se vi è una qualche regolarità nell'infinita successione dei primi. La loro distribuzione, in effetti, appare incostante, imprevedibile, quasi bizzarra. Ad esempio tra 10 e 20 vi sono ben 4 numeri primi, mentre fra 800 e 820 ce ne sono soltanto 2. Vi sono poi i cosiddetti numeri **primi gemelli**, ossia coppie di numeri primi che differiscono fra loro di 2 (ad esempio le coppie 11 e 13, 107 e 109). E' facile dimostrare inoltre che si possono trovare numeri primi consecutivi a distanza arbitrariamente grande, ossia [1, p. 38]:

Teorema. Sia $k > 1$ un intero qualunque, allora esistono k numeri naturali consecutivi nessuno dei quali è primo.

D'altra parte già nel 1845 il matematico francese Joseph Bertrand formulò l'omonima congettura (dimostrata cinque anni più tardi da Chebyshev):

Postulato di Bertrand. Per ogni intero $n > 1$ c'è sempre un numero primo compreso fra n e $2n$.

In definitiva risultava chiaro che uno dei grandi problemi legato ai numeri primi era quello di capire in che modo essi si susseguono. Una misura di questa comprensione viene epressa dalla funzione *conta-primi* $\pi(x)$ così definita:

$$\pi(x) = \text{n° dei primi } \leq x$$

Ad esempio, $\pi(10) = 4$ (i numeri primi ≤ 10 sono 2,3,5,7), $\pi(30) = 10$ e così via.
Il fatto che i numeri primi sono infiniti quindi, può essere riformulato nel modo seguente:

$$\lim_{x \to +\infty} \pi(x) = +\infty$$

Tra i vari matematici che hanno studiato la funzione $\pi(x)$, vi è senz'altro Carl Friedrich Gauss, considerato da molti il *Principe dei Matematici*. Sulla base di risultati numerici egli congetturò, nel 1792 a soli 15 anni, che $\pi(x)$ tende asintoticamente alla funzione $\frac{x}{\log x}$ per $x \to +\infty$ ($\log x$ rappresenta il logaritmo di x in base e). Cioè che per valori via via più grandi di x le funzioni $\pi(x)$ e $\frac{x}{\log x}$ assumono valori sempre più vicini (in proporzione). Sinteticamente:

$$\pi(x) \sim \frac{x}{\log x}$$

In realtà né Gauss né i suoi contemporanei riuscirono a dimostrare questa corretta intuizione. Infatti, nonostante i successivi progressi effettuati dal russo Chebyshev nella direzione di una dimostrazione rigorosa, si dovette aspettare più di un secolo affinchè, nel 1896, due eminenti matematici, Jacques Hadamard e Charles-Jean De la Vallèe-Poussin, dimostrarono, in modo indipendente, quello che è universalmente conosciuto come il

Teorema dei Numeri Primi. $\pi(x)$ è asintotica alla funzione $\frac{x}{\log x}$, ossia vale:

$$\lim_{x \to +\infty} \frac{\pi(x)}{\frac{x}{\log x}} = 1$$

4. La funzione Zeta di Riemann

Nelle rispettive dimostrazioni, sia Hadamard che De la Vallèe-Poussin utilizzarono avanzate tecniche di calcolo legate alla Teoria Analitica dei Numeri, ossia quel ramo della matematica che coniuga la Teoria dei Numeri con l'Analisi Complessa. In particolare attinsero ai preziosi risultati ottenuti, qualche decennio prima, dal grande matematico Bernard Riemann.

Questi infatti, nel suo celebre articolo del 1859 dal titolo "*Sul* numero di primi minori di una grandezza data"[1], approcciò lo studio della funzione $\pi(x)$ utilizzando metodi di analisi complessa. Più precisamente egli considerò la funzione Zeta $\zeta(s)$, definita dal Prodotto di Eulero, come una funzione di variabile complessa (tecnicamente parlando effettuò un prolungamento analitico della funzione $\zeta(s)$).

Una cosa alla volta: cos'è la funzione Zeta $\zeta(s)$ e cosa si intende per Prodotto di Eulero?

Il tutto iniziò verso il 1740 quando il matematico svizzero Eulero, uno dei più prolifici della storia, introdusse la funzione **Zeta** $\zeta(x)$ definita, per tutti i numeri reali $x > 1$, dalla somma infinita (serie):

$$\zeta(x) = \frac{1}{1^x} + \frac{1}{2^x} + \frac{1}{3^x} + \frac{1}{4^x} + \cdots = \sum_{n=1}^{+\infty} \frac{1}{n^x}$$

Per $x \le 1$ la somma dà un risultato infinito (la serie diverge) mentre per $x > 1$ la somma ritorna un valore finito (la serie converge). Eulerò dimostrò che per tali valori di x vale l'identità:

$$\zeta(x) = \sum_{n=1}^{+\infty} \frac{1}{n^x} = \prod_{p \in \mathrm{P}} \frac{1}{1 - p^{-x}}$$

nota appunto come **Prodotto di Eulero**.

La cosa sorprendente di questa identità è che il prodotto a destra è esteso a tutti i numeri primi (con P infatti si intende l'insieme di tutti i numeri primi), a conferma della natura "fondamentale" di questi ultimi.

Riemann ebbe il merito di estendere la funzione Zeta al campo dei Complessi (numeri nella forma $s = a + ib$, con $i = \sqrt{-1}$), ottenendo l'omonima **Funzione Zeta** $\zeta(s)$. Il procedimento utilizzato fu appunto quello del *prolungamento analitico* (la cui comprensione va oltre gli scopi del presente articolo) e ciò gli permise di ottenere una funzione definita su tutto il piano complesso.

L'importanza del contributo di Riemann è dovuta al fatto che vi è uno stretto legame fra la funzione $\pi(x)$ e la funzione Zeta $\zeta(s)$. Sempre in quell'articolo il matematico tedesco formulò, senza darle particolare importanza, una congettura la cui dimostrazione è ancora oggi tra i più importanti enigmi matematici:

Ipotesi di Riemann. Tutti gli zeri non banali della funzione Zeta $\zeta(s)$ hanno parte reale pari ad $\frac{1}{2}$

Agli inizi del '900 venne dimostrato che, se verificata, l'Ipotesi di Riemann avrebbe introdotto un ulteriore miglioramento nella stima dell'errore presente nel Teorema dei Numeri Primi.

La soluzione di questo problema, quindi, potrebbe guidarci ad una conoscenza più precisa della funzione $\pi(x)$ e della distribuzione dei numeri primi.

[1] Il titolo originale è *"Über die Anzahl der Primzahlen unter einer gegebenen Größe"*

5. Test di Primalità e Fattorizzazione

Oltre al problema di capire come i numeri primi si distribuiscono nell'insieme dei naturali, i matematici hanno affrontato altre avvincenti sfide. Tra queste troviamo:

- *test di primalità*: riconoscere se un intero N è primo
- *fattorizzazione*: scomporre un numero N nei suoi fattori primi

Lo stesso Gauss nel suo *Disquisitiones Arithmeticae* definì questi due problemi come i più "importanti ed utili di tutta l'aritmetica". In verità dopo i primi "elementari" algoritmi proposti dai greci, ancora nel 300 a.C, non ci furono significativi progressi fino ai contributi di Fermat, nel XVII° secolo.

La tecnica proposta dai greci era di fatto elementare: dato il numero N, si procede a dividere tale numero per 2, 3, 4, 5, …, $\sqrt{N}$. Se nessuna delle divisioni dà resto 0 allora N è primo altrimenti è composito e sono noti i suoi fattori. L'algoritmo funziona, questo è certo, ma è poco efficiente e diventa proibitivo per valori di N molto grandi.

Alla base del procedimento di fattorizzazione proposto da Fermat, invece, c'è l'idea di esprimere il numero N come differenza di due quadrati, $N = x^2 - y^2$, ottenendo così i fattori $(x - y)$ e $(x + y)$.

Dopo Fermat anche Legendre e lo stesso Gauss suggerirono altre tecniche per scomporre in fattori primi un numero intero ma alla fine rimaneva il problema che gli algoritmi individuati non erano abbastanza efficienti per fattorizzare grandi numeri (con più di 10 cifre). D'altra parte, ricordiamolo, a quell'epoca i calcoli venivano eseguiti a mano su carta.

E' per tale motivo che il problema della fattorizzazione ha suscitato un rinnovato interesse solo nel XX° secolo, con l'arrivo dei calcolatori elettronici e soprattutto a partire dagli anni '70, dopo la nascita della Crittografia a Chiave pubblica. Questo perché, come vedremo in seguito, la crittografia a chiave pubblica basa la propria sicurezza sulla difficoltà nel risolvere problemi matematici particolarmente ardui e tra questi c'è anche il problema della fattorizzazione intera.

Gli algoritmi per verificare la primalità di un numero intero hanno avuto più o meno la stessa evoluzione: i contributi più importanti si sono avuti negli ultimi decenni, in seguito all'introduzione delle moderne tecniche crittografiche.

Ciò che accomuna il problema della fattorizzazione e del test di primalità è la loro natura *computazionale*; questo perché fin dai tempi degli antichi greci siamo in grado fattorizzare un numero o verificare se è primo. Il difficile stà nel riuscire ad individuare algoritmi che risolvano tali problemi in modo "computazionalmente efficiente", cioè che non richiedano troppi calcoli e che quindi possano essere applicati anche a numeri molto grandi, fornendo il risultato in tempi *ragionevoli*.

I procedimenti più diffusi per risolvere la verifica di primalità sono gli algoritmi probabilistici di Miller-Rabin e Solovay-Strassen concepiti negli anni '70. In tempi recenti, nel 2002, i tre ricercatori indiani Agrawal, Kayena e Saxena hanno proposto l'agoritmo deterministico AKS (dalle loro iniziali) che risolve il test di primalità in modo molto efficiente (con complessità polinomiale).

Per quanto riguarda il problema della fattorizzazione, invece, lo stato dell'arte è l'algoritmo General Number Field Sieve (GNFS) sviluppato verso la fine degli anni '80, particolarmente complesso ma il più veloce in assoluto.

6. Crittografia

Dopo questa veloce incursione nel mondo nei numeri primi, è giunto il momento di capire come essi vengono applicati nella moderna crittografia. Già, ma cos'è la crittografia?

La crittografia (il termine deriva dal greco *kryptòs* - nascosto e *gràphein* - scrivere) è la scienza delle scritture segrete. Per mezzo di tecniche crittografiche quindi, un messaggio viene alterato utilizzando un procedimento concordato da *mittente* e *destinatario* in modo che risulti incomprensibile ad un eventuale *avversario* che riesca ad intercettarlo. La modifica del testo in chiaro viene detta *cifratura*, mentre il procedimento inverso, che permette di ricostruire il messaggio originale, è chiamato *decifratura*.

Come già detto, mittente e destinatario devono condividere a priori una conoscenza segreta che consenta la cifratura del messaggio e la successiva decifratura. Tale conoscenza però non è il processo di modifica ma è la cosiddetta *chiave* ossia una stringa alfanumerica che costituisce un parametro della

funzione di cifratura e della funzione di decifratura. Il metodo di alterazione perciò è noto a chiunque ma ogni volta viene parametrizzato con una chiave nota solo al mittente e al destinatario. Questo concetto è conosciuto come *Principio di Kerckhoff*, dal nome del linguista-crittografo fiammingo Auguste Kerckhoff che nel 1883 postulò tale idea in un articolo intitolato "*La cryptographie militare*":

"tutti gli algoritmi devono essere pubblici, solo le chiavi sono segrete"

Gli algoritmi crittografici possono essere suddivisi in due grandi famiglie:

- **algoritmi a chiave segreta** *(simmetrici)*: i processi di cifratura e di decifratura utilizzano la stessa chiave K.
- **algoritmi a chiave pubblica** *(asimmetrici)*: la chiave di cifratura K_E è diversa dalla chiave K_D utilizzata nel processo di decifratura. Le due chiavi sono fra loro correlate.

Le tecniche simmetriche, storicamente nate per prime, sono particolarmente veloci e robuste ma richiedono che mittente e destinatario condividano in modo sicuro la chiave K. Inoltre soffrono del *Problema della distribuzione delle chiavi*: per garantire che N individui possano comunicare in modo sicuro fra loro, dovranno essere generate $\frac{N(N-1)}{2}$ chiavi (ogni persona possiede $N-1$ chiavi). Si capisce bene che con valori di N sempre più grandi la cosa diventa complicata da gestire. Esempi di algoritmi simmetrici sono: 3DES, AES, RC4.

Il concetto di crittografia a chiave pubblica è nato negli anni '70 e più precisamente è stato proposto dai ricercatori Whitfield Diffie e Martin Hellman nel loro ormai famoso articolo "*New Directions in Cryptography*" apparso nel 1976. Il principio di funzionamento è semplice: immaginiamo che Alice desideri spedire un messaggio confidenziale a Bob:

- Bob genera due chiavi: una *privata* che custodisce gelosamente ed una corrispondente chiave *pubblica* che distribuisce a tutti i suoi *pen-friend*, tra i quali c'è anche Alice.
- Alice usa la chiave pubblica di Bob per cifrare il messaggio a lui destinato.
- Bob utilizza la propria chiave privata per decifrare il messaggio ricevuto da Alice.

La cosa importante è che con la chiave pubblica si effettua la cifratura ma con la stessa NON è possibile eseguire la corrispondente decifratura: in sostanza chi possiede la chiave pubblica può solo cifrare mentre per ricostruire il messaggio è necessario utilizzare la relativa chiave privata.

In questo modo si risolve elegantemente il problema della distribuzione delle chiavi: Bob infatti deve generare una sola coppia di chiavi e può distribuire a chi vuole la propria chiave pubblica.

Le due chiavi, pubblica e privata, sono fra loro correlate ma deve essere *difficile* risalire alla seconda conoscendo la prima. Questa difficoltà è di natura matematica, o meglio, è legata ad un problema matematico particolarmente difficile da risolvere. Ad oggi i problemi matematici su cui si basa la crittografia a chiave pubblica sono:

- **Problema della fattorizzazione intera** (IFP, *Integer Factorization Problem*): dato un numero composto n ottenuto moltiplicando due grandi numeri primi p e q ($n = pq$), trovare p e q. Su questo problema si basa l'algoritmo RSA.
- **Problema del logaritmo discreto** (DLP, *Discrete Logarithm Problem*): calcolare il logaritmo di un numero intero all'interno di un gruppo finito. Su tale problema si basano gli algoritmi El-Gamal e Diffie-Hellman.
- **Problema del logaritmo discreto su curve ellittiche** (ECDLP): consiste nella risoluzione del problema del logaritmo discreto all'interno dei punti di una curva ellittica. E' alla base di tutta la crittografia su curve ellittiche.

In questa sede analizzeremo l'algoritmo RSA, ma prima di farlo dobbiamo introdurre alcuni necessari strumenti matematici.

7. Aritmetica modulare

Gli algoritmi di cifratura agiscono su insiemi di valori che sono discreti e finiti. L'aritmetica modulare consente di effettuare le normali operazioni di somma e prodotto ottenendo risultati che sono sempre all'interno di un determinato *range* di valori. Non a caso viene anche detta *aritmetica dell'orologio*: anche se i minuti "vanno continuamente avanti" rimangono sempre all'interno dell'insieme finito e discreto 0,1,2,…,59.
In sostanza, l'aritmetica modulare all'interno di un insieme finito $Z_n = \{0,1,2,\ldots,(n-1)\}$ prevede che di ogni numero se ne consideri il *residuo modulo n,* ossia:

- si svolgono le normali operazioni di somma e moltiplicazione
- si dividono i risultati per n e se ne considerano i resti (detti anche **residui**); il resto della divisione del numero a per n si indica con $\boldsymbol{a}$ **mod** $\boldsymbol{n}$.
- il numero n viene detto **modulo**.

Consideramo alcuni esempi con l'insieme finito $Z_{11} = \{0,1,2,3,4,5,6,7,8,9,10\}$:

$6 \bmod 11 = 6$ $\quad [6 = 11\cdot 0 + \mathbf{6}]$
$(8 + 7) \bmod 11 = 15 \bmod 11 = 4$ $\quad [15 = 11\cdot 1 + \mathbf{4}]$
$(6 \cdot 4) \bmod 11 = 24 \bmod 11 = 2$ $\quad [24 = 11\cdot 2 + \mathbf{2}]$

Si possono facilmente dimostrare le seguenti proprietà:

$[(a \bmod n) \pm (b \bmod n)] \bmod n = (a \pm b) \bmod n$
$[(a \bmod n)\,(b \bmod n)] \bmod n = (ab) \bmod n$

L'aritmetica modulare inoltre definisce il concetto di *congruenza*, che è paragonabile alla relazione di uguaglianza nel caso di numeri naturali (5 = 5).

Congruenza. Due numeri a e b si dicono congruenti modulo n se vale $(a \bmod n) = (b \bmod n)$ oppure, in forma equivalente, se n divide $a - b$. In tal caso si scrive:

$$\boldsymbol{a \equiv b \pmod{n}}$$

Quindi, sempre considerando l'insieme finito $Z_{11} = \{0,1,2,3,4,5,6,7,8,9,10\}$, possiamo scrivere:

$18 \equiv 7 \pmod{11}$
$12 \equiv 1 \pmod{11}$

All'interno di insiemi finiti è anche possibile definire la proprietà di inverso:
Dato un numero a, il suo *inverso moltiplicativo* in Z_n è quel numero b per il quale vale

$$\boldsymbol{ab \equiv 1 \pmod{n}}$$

A tale riguardo vale il seguente:

Teorema. Se il numero $a \in Z_n$ è tale che $\gcd(a,n) = 1$ allora a ammette inverso moltiplicativo che è unico.

Considerando Z_{11}, si ha che $n = 11$ è primo, quindi è coprimo con tutti gli elementi non nulli di Z_{11} e pertanto questi ammettono inverso moltiplicativo.
Per esempio se prendiamo $a = 4$, si trova facilmente che il suo inverso moltiplicativo è 3: infatti vale $4\cdot 3 = 12 \equiv 1 \pmod{11}$.
L'inverso moltiplicativo in Z_n di un numero a viene calcolato per mezzo dell'Algoritmo esteso di Euclide, che nella sua forma base permette di calcolare il gcd() di due numeri.

8. Funzione di Eulero φ

Molto importante nell'ambito della Teoria dei Numeri è la funzione φ (*phi*) di Eulero, così definita:

Funzione φ di Eulero. Dato l'intero n si indica con $\varphi(n)$ il numero di elementi di Z_n che sono coprimi con n; in forma sintetica

$$\phi(n) = \#\{a \in Z_n \mid \gcd(a,n) = 1\}$$

Vediamo degli esempi:

$\varphi(5) = 4$ [5 è *coprimo con tutti gli elementi non nulli di* Z_5]
$\varphi(10) = 4$ [10 *è coprimo con i numeri* 1,3,7,9]

Abbiamo già detto che se n è primo allora tutti gli elementi di Z_n sono coprimi con n stesso, pertanto vale $\varphi(n) = n - 1$. Inoltre si può facilmente dimostrare che se $n = pq$, dove p e q sono due numeri primi, allora $\varphi(n) = \varphi(pq) = \varphi(p)\,\varphi(q) = (p-1)(q-1)$.
Ad esempio:

$\varphi(21) = \varphi(3)\,\varphi(7) = (3-1)(7-1) = 12$

Come vedremo, gioca un ruolo fondamentale nel sistema di cifratura RSA il seguente:

Teorema di Eulero. Se a ed n sono coprimi, vale la seguente relazione

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

Facciamo una verifica con i valori $a = 3$, $n = 10$ (che soddisfano $\gcd(a,n) = 1$):

$\varphi(10) = 4$
$3^4 = 81 \equiv 1 \pmod{10}$

Siamo ora in grado di analizzare l'algoritmo RSA.

9. Algoritmo RSA

Il sistema di cifratura RSA è stato proposto nel 1977 dai suoi tre autori Rivest, Shamir e Adleman. Oggi è senza dubbio l'algoritmo a chiave pubblica più utilizzato e il motivo di tanta popolarità risiede nella sua comprovata sicurezza. Sicurezza che si basa sul già citato problema della fattorizzazione intera (IFP): dato il numero n ottenuto moltiplicando due grandi numeri primi p e q ($n = pq$), trovare questi ultimi.
Per vedere come funziona consideriamo la situazione in cui Alice vuole spedire un messaggio confidenziale a Bob.

Generazione delle chiavi. Anzitutto, lo ricordiamo, Bob deve generare la coppia di chiavi pubblica e privata:

- seleziona due numeri primi p e q sufficientemente grandi
- calcola $n = pq$ e $\varphi(n) = (p-1)(q-1)$
- individua il numero e coprimo con $\varphi(n)$, ossia che verifichi $\gcd(e, \varphi(n))=1$
- calcola d tale che $de \equiv 1 \pmod{\varphi(n)}$

La **chiave privata** è la coppia di numeri $K_{pri} = (d,n)$ che viene conservata da Bob.
La **chiave pubblica** è la coppia di numeri $K_{pub} = (e,n)$ che viene distribuita a tutti, Alice compresa.
I valori e e d vengono detti esponente rispettivamente di cifratura e di decifratura.

Cifratura. Per cifrare il messaggio è necessario rappresentare quest'ultimo come un numero $m < n$. Alice quindi usa la chiave pubblica di Bob K_{pub} ed effettua la cifratura:

- calcola $c = m^e \bmod n$
- spedisce c a Bob

Decifratura. Ricevuto il testo cifrato c, Bob utilizza la propria chiave privata K_{pri} per ricostruire il messaggio originario m:

- calcola $m = c^d \bmod n$

In breve, durante la cifratura il messaggio m viene elevato ad e (modulo n) mentre in ricezione il testo cifrato c viene elevato a d (modulo n).

Dimostrazione

Per capire come funziona l'algoritmo RSA evidenziamo quanto segue:

- l'esponente di cifratura è stato scelto coprimo con $\varphi(n)$, in modo da ammettere inverso moltiplicativo modulo $\varphi(n)$.
- l'esponente d è proprio l'inverso moltiplicativo di e, cioè tale che $de \equiv 1 \pmod{\varphi(n)}$. Questo significa che i due esponenti soddisfano la relazione $de = k\varphi(n) + 1$, con k intero.

Vediamo ora cosa succede durante la decifratura:

$c^d \bmod n = [(m^e \bmod n)^d] \bmod n = m^{ed} \bmod n = m^{(k\varphi(n)+1)} \bmod n =$

$= [(m^{\varphi(n)})^k m] \bmod n = [(m^{\varphi(n)})^k \bmod n)\ (m \bmod n)] \bmod n = [((m^{\varphi(n)})^k \bmod n)\ m] \bmod n$ (1)

applicando il Teorema di Eulero(*), sviluppiamo il primo fattore dell'ultimo passaggio della (1) :

$(m^{\varphi(n)})^k \bmod n = [(m^{\varphi(n)} \bmod n)^k] \bmod n = 1^k \bmod n = 1$ (2)

quindi, inserendo la (2) nella (1), otteniamo proprio il messaggio originario:

$c^d \bmod n = \ldots\ldots = [((m^{\varphi(n)})^k \bmod n)\ m] \bmod n = [1 \cdot m] \bmod n = m$

(*) Al lettore più attento non sarà sfuggito che nella (2) si è applicato il Teorema di Eulero in modo improprio, perché non abbiamo fatto alcuna ipotesi sulla coprimalità fra m ed n (come richiederebbe il Teorema stesso). In realtà si può facilmente verificare che, siccome $m < n$ e quest'ultimo è uguale al prodotto di due numeri primi p e q, la dimostrazione appena vista è corretta anche per le situazioni in cui $\gcd(m,n) \neq 1$ [7, p.178].

Riepiloghiamo con un esempio:

- si considerino i primi $p = 11$ e $q = 17$, quindi $n = pq = 187$.
- calcoliamo $\varphi(187) = 160$ e i due esponenti $e = 7$, $d = 23$
- le chiavi di Bob sono: $K_{pri} = (23, 187)$ e $K_{pub} = (7, 187)$
- rappresentiamo il messaggio con il numero $m = 88$
- Alice invia il messaggio cifrato: $c = m^e \bmod n = 88^7 \bmod 187 = 11$
- Bob ricostruisce m: $c^d \bmod n = 11^{23} \bmod 187 = 88$

Sicurezza

Per un attaccante violare il sistema RSA significa recuperare il messaggio originario m a partire dal testo cifrato c. Attualmente l'unico modo conosciuto per fare questo utilizza l'esponente di decifratura d. Anche se, bisogna dirlo, non è stato dimostrato che la decifratura di c richiede necessariamente la conoscenza dell'esponente d.

In ogni caso l'attacco più ovvio consiste nel fattorizzare n nei primi p e q: così facendo è possibile calcolare $\varphi(n) = (p-1)(q-1)$ e quindi ottenere l'inverso moltiplicativo del valore e, ossia d.

Ma la fattorizzazione di n è il solo modo per "rompere" l'algoritmo RSA? Ossia è l'unico modo per recuperare l'esponente d ? In verità se si riuscisse a calcolare direttamente il valore $\varphi(n)$, senza fattorizzare n, sarebbe comunque immediato ottenere l'esponente d. Inoltre possiamo dimostrare che la conoscenza di $\varphi(n)$ permette di calcolare agevolmente i primi p e q:

- si osservi che: $\varphi(n) = (p-1)(q-1) = pq - p - q + 1 = n - (p+q) + 1$
- i valori p e q sono le due radici dell'equazione: $x^2 - (p+q)x + pq = 0$ (3)
- riscrivendo la (3) utilizzando $\varphi(n)$ ed n si ottiene: $x^2 - (n - \varphi(n) + 1)x + n = 0$
- di conseguenza, noti n e $\varphi(n)$ è immediato ricavare p e q.

Quindi si può dire che calcolare $\varphi(n)$ non è più facile della fattorizzazione di n, anzi, estendendo la considerazione, possiamo enunciare la seguente

Congettura RSA. Il problema di violare il sistema RSA è difficile quanto la fattorizzazione di n.

Una dimostrazione rigorosa di tale congettura non esiste ma è opinione condivisa che essa sia vera. Per realizzare un sistema RSA sicuro, quindi, il numero $n = pq$ dovrà essere abbastanza grande da rendere computazionalmente impraticabile la propria fattorizzazione. Oggi questo significa che n deve avere tra le 300 e 600 cifre decimali!

10. Conclusioni

La crittografia è una materia vasta, certamente, e insieme all'RSA vengono utilizzati numerosi altri algoritmi, sia simmetrici che asimmetrici. In ogni caso è bene sapere che quando, ad esempio, utilizziamo la nostra carta di credito per acquistare su Amazon piuttosto che per fare la spesa al supermercato noi *usiamo i numeri primi*.

Bibliografia

Un libro che spiega molto bene la Teoria dei Numeri è [2] mentre [1] descrive il legame esistente tra quest'ultima e la moderna crittografia. Nei testi [3] e [4] è possibile trovare, con numerose note storiche, un'accessibile introduzione alla funzione Zeta di Riemann e alla relativa Ipotesi. Un vero riferimento sullo studio dei numeri primi, soprattutto per gli aspetti computazionali, è [5]. I libri [6] e [7] sono due tradizionali testi di crittografia, utilizzati in corsi universitari: il primo è più attento alle applicazioni informatiche mentre il secondo cura maggiormente gli aspetti teorici degli algoritmi. Infine [8] propone una godibilissima e avvincente storia della crittografia.

[1] Leonesi S., Toffalori C., *Numeri e Crittografia*, Springer (2006)
[2] Jones G.A., Jones J.M., *Elementary Number Theory*, Springer (2005)
[3] Derbyshire J., *L'ossessione dei numeri primi*, Bollati Boringhieri (2006)
[4] Du Sautoy M., *L'enigma dei numeri primi*, BUR (2005)
[5] Crandall R., Pomerance C., *Prime numbers, a computational perspective*, Springer (2005)
[6] Stallings W., *Crittografia e Sicurezza delle Reti*, Mc Graw-Hill (2003)
[7] Paar C., Pelzl J., *Understanding Cryptography*, Springer (2010)
[8] Singh S., *Codici e segreti*, BUR (2001)