Aritmetica e Crittografia

Luigi Ambrosio

Scuola Normale Superiore, Pisa
http://cvgmt.sns.it

SCUOLA
NORMALE
SUPERIORE

Indice

1. Il problema della comunicazione sicura
   - Un possibile schema di comunicazione
   - Il codice di Vigenére
   - Una versione aritmetica dello schema di comunicazione
   - Il problema dello scambio delle chiavi
2. L'aritmetica dell'orologio
3. L'algoritmo di Euclide
   - Calcolo di $a^{-1} \mod (n)$ con l'algoritmo di Euclide
   - Costo computazionale dell'algoritmo di Euclide
4. Il protocollo di Rivest, Shamir e Adleman

SCUOLA NORMALE SUPERIORE

Il problema della comunicazione sicura

A deve comunicare a **B** un messaggio riservato.

Deve quindi farlo in modo che nessun terzo agente **C**, quand'anche venisse in possesso del messaggio, sia in grado di carpirne il contenuto.

Per far questo, il messaggio viene manipolato da **A** in modo tale che sia inintellegibile a terzi. Questo primo processo è detto *crittazione*.

Il processo inverso, di ricostruzione da parte di **B** del messaggio manipolato, è detto *decrittazione*.

La *crittografia* è lo studio dei metodi di crittazione e di decrittazione più efficaci.

Bibliografia

[1] G.ALBERTI: "*Aritmetica finita e crittografia a chiave pubblica: un percorso didattico per studenti delle Scuole Medie Superiori*". In: "Ricordando Franco Conti", Edizioni della Scuola Normale Superiore, 2004.

[2] N.KOBLITZ: "*A Course in Number Theory and Cryptography*". Graduate texts in Mathematics **114**, Springer, New York, 1987.

[3] S.SINGH: "*The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*", Anchor Books, Londra, 1999. Traduzione italiana: "*Codici & Segreti*", Rizzoli, Milano, 1999.

Un possibile schema di comunicazione

- **A** divide il testo in blocchi di *m* caratteri;
- poi sostituisce ciascun blocco con un altro, sempre di *m* caratteri, usando un opportuno “dizionario”;
- **B**, che è in possesso di copia del dizionario, ricevuto il messaggio crittato da **A**, compie il processo inverso e decritta il messaggio.

Esempio (Il codice di sostituzione cesareo)

`a`$\to$`c, b`$\to$`d, c`$\to$`e`,...., fino a `v`$\to$`a, z`$\to$`b`
Quindi, `codice` diventa... `eqfmeh`

Questa procedura presenta tuttavia parecchie limitazioni...
Questo ci porta al concetto di *chiave* di un codice.

Il codice di Vigenère

Questo codice, ritenuto indecifrabile fino alla metà del XIX secolo, è una sorta di codice cesareo a geometria variabile.
Supponiamo che la chiave sia 5873. Allora si cambia il primo carattere del messaggio con quello 5 posti a destra, il secondo con quello 8 posti a destra, e così via, ripetendo ciclicamente lo schema.

Quindi, `codice cesareo` diventa... `hzmnho lhaibht`

Il vantaggio è che non è necessario che **A** e **B** condividano alcun vocabolario, ma semplicemente la sequenza 5873, detta *chiave di crittazione/decrittazione.*
Anche questa procedura non è tuttavia esente da rischi: nel momento in cui **A** e **B** si scambiano la chiave, questa comunicazione (necessariamente non crittata) potrebbe essere intercettata da **C**.

Alcune date

- ’800. Primi metodi di decrittazione dei codici di sostituzione basati sull’analisi delle frequenze
- 1580. Vigenère sviluppa il suo codice
- 1850. Babbage raffina il metodo dell’analisi delle frequenze per decifrare anche il codice di Vigenère
- 1900-1940. Sviluppo di codici sempre più complessi basati su sostituzione/transposizione
- 1975. Diffie, Helmann, Merkle sviluppano il concetto di codice a chiave pubblica
- 1977. Rivest, Shamir e Adleman inventano il codice RSA

Il problema dello scambio delle chiavi

È possibile individuare una procedura di comunicazione sicura che non implichi *alcuno* scambio preventivo di chiavi tra **A** e **B** ?
Sembrerebbe impossibile, ma non è cosí :

- **A** chiude il suo messaggio una scatola, la sigilla con un lucchetto e la spedisce a **B**
- **B** si limita a rispedire la scatola (che non potrebbe aprire) a **A**, aggiungendovi però un suo lucchetto
- **A** rimuove il suo lucchetto e spedisce la scatola (ancora con il lucchetto di **B**) a **B**, che può finalmente aprirla.

Questa piccola storia mostra che la comunicazione sicura è possibile anche *senza* uno scambio preventivo di chiavi !

Una versione aritmetica dello schema di comunicazione

Fissiamo un intero $n > 1$ e definiamo

$$\mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}\,.$$

Possiamo allora tradurre in termini aritmetici lo schema di crittazione/decrittazione, considerando la chiave di crittazione come una opportuna funzione $f : \mathbb{Z}_n \to \mathbb{Z}_n$.

▸ Il concetto di funzione

- **A** converte il messaggio in una sequenza di numeri x_1, x_2, x_3,... in $\mathbb{Z}$ (ad esempio il codice ASCII, con $n = 127$);
- poi trasmette a **B** la sequenza di numeri nella sequenza y_1, y_2, y_3,... con $y_1 = f(x_1)$, $y_2 = f(x_2)$, $y_3 = f(x_3)$,...;
- infine, **B** usa la funzione inversa $f^{-1} : \mathbb{Z}_n \to \mathbb{Z}_n$ per ricostruire la sequenza iniziale x_1, x_2, x_3,... e quindi il messaggio iniziale.

Il codice cesareo rientra in questa procedura: basta prendere $n = 21$, dimodoché $\mathbb{Z}_n$ codifica l'alfabeto italiano, e $f(a) = a + 1$, pur di convenire che $f(20) = 0$.

Il problema dello scambio delle chiavi

Per ovviare a questo problema **B** spedisce a **A** una descrizione della *chiave di crittazione* f, ovviamente in maniera non crittata. Questa comunicazione è intercettabile da tutti, quindi f è anche detta *chiave pubblica.*
Poi, **A** opera come si è detto in precedenza e **B**, ricevuto il messaggio crittato, lo decodifica usando la *chiave di decrittazione* f^{-1}, solo a lui nota, e per questo detta *chiave privata.*

Il segreto è, quindi, quello di cercare funzioni f per le quali il calcolo di f^{-1} sia nella pratica molto difficile, se non impossibile, per chi conosca solo f.

Per analogia, si pensi a f come alla funzione che a una parola in italiano associa una parola dello stesso significato in giapponese....

L'aritmetica dell'orologio

È naturale cercare funzioni $f : \mathbb{Z}_n \to \mathbb{Z}_n$ "difficili da invertire" di tipo aritmetico. Quindi, vogliamo definire le operazioni di somma e prodotto in $\mathbb{Z}_n$.

Definizione (Congruenze modulo n)

Diremo che a e b sono congrui modulo n, e scriveremo

$$a \equiv b \bmod (n)$$

se $a - b$ è un multiplo di n.

Ad esempio, 3 e 15 sono congrui modulo 12, 22 e 7 sono congrui modulo 5, etc.

Definizione (Riduzione modulo n)

Se divido m per n, e ottengo $m = qn + r$, allora $m \equiv r \bmod (n)$. Il numero r è detto riduzione di m modulo n.

L'aritmetica dell'orologio

Definizione (Somma e prodotto in $\mathbb{Z}_n$)

Dati a, b interi, indichiamo con

$$a + b \bmod (n), \qquad a \cdot b \bmod (n)$$

le riduzioni rispettivamente della somma $a + b$ e del prodotto $a \cdot b$.

Questa "nuova" aritmetica è nota come *aritmetica modulare*. Quindi:

$$\begin{aligned}
7 + 6 = 13 &\equiv 3 && \bmod (10) \\
75 + 96 = 171 &\equiv 71 && \bmod (100) \\
115 + 120 = 235 &\equiv 4 && \bmod (11) \\
5 \cdot 4 = 20 &\equiv 2 && \bmod (6) \\
115 \cdot 120 = 13800 &\equiv 6 && \bmod (11) \\
2 \cdot 4 = 8 &\equiv 1 && \bmod (7)
\end{aligned}$$

Somma e prodotto in $\mathbb{Z}_n$ godono delle usuali proprietà (associativa, commutativa, distributiva).

Opposto e inverso in $\mathbb{Z}_n$.

È possibile definire, in $\mathbb{Z}_n$, l'opposto $-a$ di a e l'inverso a^{-1} di a:

$$-a + a = 0 \bmod (n) \qquad a^{-1} \cdot a = 1 \bmod (n).$$

Esempio

Se $n = 17$, l'opposto di $5 \bmod (n)$ vale 12, mentre l'inverso di $5 \bmod (n)$ vale 7, dato che

$$5 \cdot 7 = 35 = 1 + 2 \cdot 17 \equiv 1 \bmod (17).$$

Non è sempre possibile definire l'inverso, ad esempio se a e n sono pari (perché ?). Tuttavia, questo è sempre possibile se n è primo: basta osservare che i resti della divisione per n degli n multipli di a

$$a,\ 2a,\ 3a, \ldots, na$$

sono tutti diversi tra loro, quindi almeno uno di questi vale 1.

Un possibile codice a chiave pubblica

Un possibile codice potrebbe essere legato alla chiave pubblica (di crittazione)

$$f(x) = a \cdot x + b \bmod (n)$$

e a quella privata (di decrittazione)

$$f^{-1}(y) = a^{-1} \cdot (-b + y) \bmod (n)$$

La "segretezza" del codice dipende dalla difficoltà di ricavare a^{-1}, noto a.
Cerchiamo di precisare meglio questa "difficoltà" con il concetto di *costo computazionale di un algoritmo*.

Costo computazionale di un algoritmo

Definizione

Il costo computazionale di un'algoritmo è il numero (massimo) delle operazioni elementari necessarie per eseguirlo.

Cifre decimali di un numero n: $\log n$ (esempio, $n = 1347$, $\log n = 3, \ldots$).

Esempio

Il costo computazionale del prodotto $n \cdot m$ è dell'ordine di $\log n \cdot \log m$.

Useremo la notazione $\mathcal{O}(p)$ per indicare "dell'ordine di p"

Costo computazionale del calcolo dell'inverso $\mod(n)$.

Il costo computazionale del calcolo di a^{-1} sembrerebbe quindi

$$\text{numero dei tentativi} \times \mathcal{O}(\log^2 n) = \mathcal{O}(n \log^2 n),$$

molto maggiore del costo di cifratura:

$$\text{lunghezza del messaggio} \times \mathcal{O}(\log^2 n).$$

Con $n \sim 10^{50}$, saremmo al di fuori della portata di qualsiasi computer esistente. Ma le cose non stanno affatto cosí.

L'algoritmo di Euclide

L'algoritmo di Euclide per il calcolo del massimo comun divisore $\mathrm{MCD}(a;b)$ di a e b, con $a > b$, si basa su una semplice osservazione:

Se r è il resto della divisione di a per b (*i.e.* $a = qb + a'$, con q intero e $0 \leq a' < b$), allora $\mathrm{MCD}(a;b) = \mathrm{MCD}(a';b)$.

$$\begin{aligned} 560 : 427 &= 1 \quad \text{con resto } 133 \\ 427 : 133 &= 3 \quad \text{con resto } 28 \\ 133 : 28 &= 4 \quad \text{con resto } 21 \\ 28 : 21 &= 1 \quad \text{con resto } 7 \\ 21 : 7 &= 3 \quad \text{con resto } 0 \end{aligned}$$

Quindi $\mathrm{MCD}(560;427) = 7$ (in 5 passi).

Calcolo di $a^{-1} \mod(n)$ con l'algoritmo di Euclide

Un sottoprodotto dell'algoritmo di Euclide per il calcolo di $\mathrm{MCD}(a;b)$ è la possibilità di trovare interi relativi c_1 e c_2 tali che

$$c_1 a + c_2 b = \mathrm{MCD}(a;b).$$

Per far questo, basta ripercorrere "a ritroso" il procedimento: nel caso dell'esempio precedente abbiamo

$$\begin{aligned}
\mathrm{MCD}(560;427) &= 7 \\
&= 28 - 1 \cdot 21 \\
&= 28 - 1 \cdot (133 - 4 \cdot 28) = -133 + 5 \cdot 28 \\
&= -133 + 5 \cdot (427 - 3 \cdot 133) = 5 \cdot 427 - 16 \cdot 33 \\
&= 5 \cdot 427 - 16 \cdot (560 - 1 \cdot 427) = -16 \cdot 560 + 21 \cdot 427.
\end{aligned}$$

In particolare, se $b = n$ e $\mathrm{MCD}(a;n) = 1$ la formula precedente ci dà

$c_1 a + c_2 n = 1$, quindi... c_1 è l'inverso di $a \mod(n)$!

Numero massimo di passi necessario

Quanti passi p sono necessari, al massimo, perché l'algoritmo si concluda ?

Primo passo: $a = qb + a' \geq b + a' > 2a'$.
Secondo passo: $b = q'a' + b' \geq a' + b' > 2b'$.

Quindi, i valori della coppia $\mathrm{MCD}\,(a; b)$ si sono almeno dimezzati ogni due passi. Se quindi $\max\{a, b\} \leq 2^k$ serviranno al più $2k$ passi.

$k \leq \log_2 n + 1,$ quindi... $\qquad p \leq 2 + 2\frac{\log n}{\log 2}.$

Il costo computazionale dell'algoritmo di Euclide risulta quindi $\mathcal{O}(\log n)$, e quello per l'inverso $\mathcal{O}(\log^3 n)$, appena maggiore del costo di crittazione $\mathcal{O}(\log^2 n)$.

Elevamento a potenza in $\mathbb{Z}_n$

Abbiamo visto che le operazioni aritmetiche di base non sono adeguate per un codice a chiave pubblica. Quindi è naturale considerare operazioni più complesse, come l'elevamento a potenza.

Teorema (Fermat, 1630)

Se n è un numero primo e a soddisfa $\mathrm{MCD}(a; n-1) = 1$, allora la funzione $f(x) = x^a \bmod (n)$ è invertibile in $\mathbb{Z}_n$ e la sua inversa è

$$f^{-1}(y) = y^b \bmod (n),$$

ove b è un qualsiasi intero tale che $ab = 1 \bmod (n-1)$.

Il protocollo RSA

Il protocollo di comunicazione di Rivest, Shamir e Adleman è basato su questi ingredienti:

- Un numero n, prodotto di due (grandi) numeri primi p e q.
- Un intero a tale che $\mathrm{MCD}\,(a;(p-1)(q-1)) = 1$ e $a \leq n$.
- Un intero b tale che $ab \equiv 1 \bmod ((p-1)(q-1)) = 1$.

La chiave pubblica di crittazione è

$$f(x) = x^a \bmod (n)$$

La chiave privata di decrittazione è

$$g(y) = y^b \bmod (n)$$

Perché il protocollo funziona ? Perché un'opportuna estensione del teorema di Fermat ci garantisce che $g(y)$ è effettivamente l'inverso di $f(x)$.

Sicurezza del protocollo RSA

$n \sim 10^{300}$, h=1000, $h \log^2 n \sim 9 \cdot 10^7$

Progressi nella fattorizzazione di grandi numeri:

45 cifre, anni ’70, 70 cifre, anni ’80, 150 cifre anni ’90

Nuovi algoritmi di fattorizzazione ? Per un nuovo test di primalità (Agrawal, Kayal, Saxena, 2002), vedi

http://mathworld.wolfram.com/news/2002-08-07/primetest/

- algoritmi “quantistici” (Shor, 1994)
- Crittografia quantistica

La lotta tra crittatori e decrittatori continua....