Appunti di Sicurezza delle reti

Lorenzo Prosseda, a.a. 2018-2019

Indice

Parte 1. Crittografia 5

Capitolo 1. Teoria dei numeri 7
- 1.1. Proprietà degli interi 7
- 1.2. Numeri primi 7
- 1.3. Algoritmo di Euclide 8

Parte 2. Protocolli e sistemi per la comunicazione sicura 11

Appendice A. Introduzione alla crittografia 13

Indice analitico 17

Parte 1

Crittografia

CAPITOLO 1

Teoria dei numeri

1.1. Proprietà degli interi

Da ora in avanti parleremo di numeri interi, positivi o negativi, operando all'interno dell'insieme $\mathbb{Z}$; enunciamo la proprietà di divisione nel modo seguente: presi due interi a, $b \in \mathbb{Z}$ non uguali ($a \neq b$) si dice che a divide b quando a è un divisore di b, ovvero

$$a \backslash b \implies \exists k \, : \, b = k \cdot a$$

Dalla precedente deduciamo che b dovrà essere un multiplo di a. Inoltre, otteniamo anche che:

$$\forall a \in \mathbb{Z} \, : \, a \backslash 0; \; \nexists a \in \mathbb{Z} \, : \, 0 \backslash a; \; \forall a \in \mathbb{Z} \, : \, a \backslash a$$

La relazione di divisione introdotta ammette la proprietà transitiva:

$$\forall a, b, c \in \mathbb{Z} \wedge a \neq b \neq c \, : \, a \backslash b \wedge b \backslash c \implies a \backslash c$$

1.2. Numeri primi

1.2.1. Definizione e proprietà. Un numero a si dice primo quando è divisibile solo per 1 e per sé stesso (ovvero se vale $\forall b \in \mathbb{Z} \, : \, b \backslash a \iff b = 1 \vee b = a$); un numero composto è scomponibile in un numero finito di fattori, e questa scomposizione è unica. Determinare la primalità di un numero tuttavia non è cosa facile; introduciamo il seguente teorema:

Teorema dei numeri primi

Teorema 1.1. *Sia $\pi(n)$:="numero di numeri primi fino a n", allora vale*

$$\pi(n) \sim \frac{n}{\ln(n)}$$

Alcuni algoritmi crittografici usano i numeri primi come "ingredienti" per creare le chiavi: in questi casi il teorema introdotto si dimostra molto utile per determinare la quantità di numeri primi che è possibile ottenere con una data quantità di cifre.

Esempio 1.1. Determinare una stima della quantità di numeri primi che è possibile ottenere a partire da 100 cifre.

✓Usando il Teorema 1.1 possiamo scrivere la quantità di numeri primi con 100 cifre come

$$\pi\left(10^{100}\right) - \pi\left(10^{99}\right) = \frac{10^{100}}{100 \ln(10)} - \frac{10^{99}}{99 \ln(10)} \simeq \boxed{10^{97}}$$

□

Se fossimo nel contesto di un algoritmo di cifratura, pur sapendo che la chiave sia un numero primo di 100 cifre, dovremmo analizzare in ogni caso 10^{97} possibili candidati.

Prendiamo ora a, $b \in \mathbb{Z}$ tali che $\text{MCD}(a, b) = 1$: in tal caso diremo che e e b sono *coprimi* o *primi relativi*, indicando la loro relazione come $a \perp b$; da questa relazione segue che due numeri coprimi non hanno fattori in comune.

La funzione $\text{MCD}(a, b)$ indica il massimo comune divisore tra a e b

Escluso il numero 2, tutti i primi sono dispari, e sono divisi in due classi: preso un numero primo p, esso appartiene a una delle seguenti classi:

- $p \equiv 1 \pmod 4$
- $p \equiv 3 \pmod 4$

L'operatore $\equiv$ indica la congruenza in modulo: si dice che un numero $a \in \mathbb{Z}$ è *congruente a 1 modulo n* (e si scrive $a \equiv 1 \pmod n$) se il resto della divisione di a per n è 1.

Componendo le due classi osserviamo che tutti i numeri primi p possono essere indicati come

$$p = 6k \pm 1 \implies p \equiv \pm 1 \pmod 6, \; k \in \mathbb{Z} \tag{1.2.1}$$

Troveremo per esempio $p_{k=1} = 6 \pm 1 = \{5, 7\}$, $p_{k=2} = 12 \pm 1 = \{11, 13\}$, $p_{k=3} = 18 \pm 1 = \{17, 19\}, \ldots$; osserviamo che tutti i numeri della successione appena definita sono primi, tuttavia non tutti i primi appartengono a questa successione.

1.2.2. Test di primalità con classi. Possiamo usare la successione (1.2.1) per testare la primalità di un numero intero: sia $n > 0$ un numero del quale si vuole conoscere la primalità; allora definiamo un algoritmo iterativo che costruisce la successione (1.2.1) incrementando k, e per ogni primo p_k ottenuto, se non vale $p_k \backslash n$ fino a che $6k + 1 \leq \sqrt{n}$, allora n è primo.

1.3. Algoritmo di Euclide

1.3.1. Definizione della successione. La funzione principale di questo algoritmo, è calcolare il massimo comune divisore di due numeri; presi due interi m e n tali che $m < n$, vogliamo calcolare $\textsc{mcd}\,(m,\, n)$. Tramite questo algoritmo otteniamo il risultato desiderato, senza passare per la scomposizione in fattori primi di m e n; definendo la seguente successione:

$$\begin{gathered} \textsc{mcd}\,(m,\, n) = \textsc{mcd}\,(n \bmod m,\, n) = \textsc{mcd}\,((n \bmod m) \bmod n,\, n) = \\ \ldots = \textsc{mcd}\,(0,\, n) = n \end{gathered} \tag{1.3.1}$$

Osserviamo come calcolare questa successione con un esempio.

Esempio 1.2. *Calcolare il massimo comune divisore tra 482 e 1180*

✓Procediamo applicando la definizione (1.3.1): per farlo dovremo scomporre il numero maggiore (n dalla definizione) usando il suo modulo rispetto al minore (m); in pratica prendiamo 1180 e lo dividiamo per 482, conservando il residuo dell'operazione da usare nel termine successivo della successione

$$\textsc{mcd}\,(482,\, 1180) \to 1180 = 2 \cdot 482 + 216$$

proseguiamo lavorando col resto del passo precedente (216) e col valore precedentemente usato per calcolare il modulo (482)

$$\begin{gathered} \textsc{mcd}\,(216,\, 482) \to 482 = 2 \cdot 216 + 50 \\ \textsc{mcd}\,(50,\, 216) \to 216 = 4 \cdot 50 + 16 \\ \textsc{mcd}\,(16,\, 50) \to 3 \cdot 16 + \mathbf{2} = \boldsymbol{d} \\ \textsc{mcd}\,(2,\, 16) \to 8 \cdot 2 + \boxed{0} \end{gathered}$$

La successione termina quando si ottiene resto zero; il resto chiamato d, ottenuto alla penultima riga (sopra a quella con resto 0, vale 2 in questo caso) è effettivamente il risultato della richiesta: $\textsc{mcd}\,(482,\, 1180) = 2$ □

1.3.2. Algoritmo in forma simbolica. Presi $a, b \in \mathbb{Z}$ tali che $a < b$, otteniamo $\textsc{mcd}\,(a, b) = d$ applicando (1.3.1) nel seguente modo:

Algoritmo di Euclide

$$\begin{aligned} b &= q_1 \cdot a + r_1 \\ a &= q_2 \cdot r_1 + r_2 \\ r_1 &= q_3 \cdot r_2 + r_3 \\ &\vdots \\ r_{k-2} &= q_k \cdot r_{k-1} + r_k \\ r_{k-1} &= q_{k+1} \cdot r_k + 0 \\ &\boxed{r_k = d} \end{aligned} \tag{1.3.2}$$

dove q_i è l'i-esimo quoziente e r_i è l'i-esimo resto; dall'algoritmo si può dedurre che, presi $a, b \neq 0$ e sia $d = \textsc{mcd}\,(a, b)$, allora è vero che $\exists x, y \in \mathbb{Z} \, : \, a \cdot x + b \cdot y = d$. Questo risulta chiaro se immaginiamo che i due interi cercati siano anche negativi; per trovare tali interi è necessario utilizzare una estensione del (1.3.2).

1.3.3. Algoritmo esteso. L'algoritmo di Euclide presentato nella sottosezione precedente può essere esteso, impiegando nel suo svolgimento due successioni $\{x_k\}$ e $\{y_k\}$: esse avranno i primi due valori ben definiti, come

$$x_0 =, \, x_1 = 1; \; y_0 = 1, \, y_1 = 0 \tag{1.3.3}$$

Facendo corrispondere ai passi (1.3.2) gli elementi delle successioni $\{x_k\}$ e $\{y_k\}$, possiamo ottenere gli elementi dal secondo in poi come:

Algoritmo di Euclide esteso

$$\begin{aligned} x_2 &= -q_1 \cdot x_1 + x_0 & y_2 &= -q_1 \cdot y_1 + y_0 \\ x_3 &= -q_2 \cdot x_2 + x_1 & y_3 &= -q_2 \cdot y_2 + y_1 \\ x_4 &= -q_3 \cdot x_3 + x_2 & y_4 &= -q_3 \cdot y_3 + y_2 \\ &\vdots & &\vdots \\ \boxed{x_{k+1}} &= -q_k \cdot x_k + x_{k-1} & \boxed{y_{k+1}} &= -q_k \cdot y_k + y_{k-1} \end{aligned} \tag{1.3.4}$$

Osserviamo che gli elementi delle successioni in x e y si ottengono in modo analogo, tuttavia le due successioni sono inizializzate in modo differente (1.3.3). Euclide Esteso ci permette di affermare che

$$\textsc{mcd}\,(a, b) = d = a \cdot x_{k+1} + b \cdot y_{k+1}$$

dove gli interi x_{k+1} e y_{k+1} sono ottenuti dalle successioni (1.3.4); normalmente uno dei due è positivo e l'altro è negativo.

Parte 2

Protocolli e sistemi per la comunicazione sicura

APPENDICE A

Introduzione alla crittografia

Cenni storici. Il termine crittografia deriva dal greco $\kappa\rho\upsilon\pi\tau\acute{o}\zeta$ (kryptós - segreto) e $\gamma\rho\alpha\varphi\acute{\eta}$ (grafi - scrittura); tra gli esempi di crittografia dal passato il più famoso è il cifrario di Cesare: si tratta di un cifrario a scorrimento ciclico, che consisteva nello scrivere le lettere dell'alfabeto su due anelli per poi ruotarne uno rispetto all'altro di $k = +3$ posizioni; in questo modo si ottiene $A \to D$, $B \to E$, $\ldots Z \to C$.

Non si trattava di un cifrario robusto, ma veniva usato quando gli avversari dell'Impero Romano erano i Galli: si rivelò un metodo più che sufficiente.

Comunicazione sicura. In generale una comunicazione sicura tra due parti si svolge nel modo seguente: sia A il mittente del messaggio, B il destinatario, ed E un intruso che abbia accesso al canale di comunicazione. L'intruso può essere *passivo* (intercetta i messaggi senza modificarne il flusso) oppure *attivo* (modifica il contenuto dei messaggi).

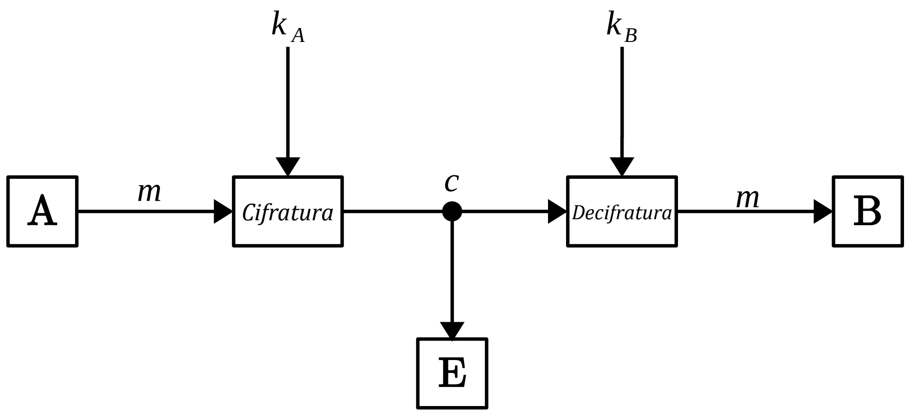

Figura A.0.1. Scenario fondamentale di comunicazione

A invia un *plaintext* m (testo in chiaro), lo codifica usando una funzione di cifratura (un algoritmo crittografico) che prende in ingresso anche la sua chiave k_{A} e ottiene un *ciphertext* c (messaggio cifrato); il messaggio c giunge a ***B***, il quale usa una funzione di de-cifratura — tramite la sua chiave k_{B}, che è associata in qualche modo alla chiave k_{A} — per ottenere nuovamente il *plaintext* m inviato da ***A***.

E potrebbe avere le seguenti intenzioni malevole rispetto alla comunicazione tra ***A*** e ***B***:

- leggere il messaggio e comprenderne il contenuto;
- ottenere la chiave;
- corrompere il contenuto del messaggio;
- impersonare A senza che B se ne accorga.

L'intruso può mettere in atto i seguenti tipi di attacchi sull'algoritmo di cifratura usato nella comunicazione:

- CIPHERTEXT-ONLY: avendo a disposizione il testo cifrato, si cerca di ricavarne delle informazioni (attacco più comune);

- KNOWN PLAINTEXT: avendo a disposizione una coppia di testo cifrato e testo in chiaro corrispondente, si confrontano i due cercando di ottenere informazioni sulla chiave;
- CHOSEN PLAINTEXT: avendo a disposizione la stessa implementazione dell'algoritmo utilizzato per cifrare il messaggio, si scelgono dei testi in chiaro da cifrare e si osservano i testi cifrati in uscita, per cercare di ricavare informazioni sull'implementazione;
- CHOSEN CIPHERTEXT: avendo a disposizione la stessa implementazione dell'algoritmo utilizzato per decifrare il messaggio, si scelgono dei testi cifrati da decifrare e si osservano i testi in chiaro in uscita, per cercare di ricavare informazioni sull'implementazione.

Sicurezza e segretezza. Giulio Cesare basava la sicurezza del proprio algoritmo di cifratura sul fatto che i possibili avversari non ne conoscessero il funzionamento; il crittografo olandese Auguste Kerckhoffs, enunciò ne *'La cryptographie militaire'* (1883) il principio di Kerchoffs:

Principio di Kerchoffs

La sicurezza di un sistema di cifratura è basata sulla segretezza della chiave (assumere sempre che il nemico conosca l'algoritmo di cifratura)

Da questa considerazione segue che la chiave utilizzata deve essere lunga, complessa, e in generale essere costruita per evitare che sia possibile indovinarla.

Claude Elwood Shannon, che scrisse cinque articoli che cambiarono la storia della comunicazione dell'informazione, tra cui un articolo sulla crittografia[1], espresse lo stesso principio in maniera molto incisiva con le parole "Il nemico conosce il sistema" (frase nota come massima di Shannon).

È interessante notare come si è passati dal fondare la sicurezza del sistema sulla segretezza dell'algoritmo alla segretezza della chiave; il passo successivo fu il sistema a *chiave pubblica*: gli algoritmi usati sono noti e accessibili a tutti, e una chiave del mittente (quella pubblica) è resa nota a tutti; tramite la chiave pubblica è possibile cifrare i messaggi, tuttavia la chiave per decifrare, associata alla chiave pubblica, è mantenuta riservata (si parla di *chiave privata*).

Fondamentale è il fatto che, non ostante esista una regola (formula o algoritmo) che permetta di associare la chiave pubblica a quella privata, per un intruso qualunque è impossibile, dal punto di vista computazionale, risalire alla chiave privata attraverso quella pubblica. Solo il mittente che possiede la chiave privata è in grado di computare questa associazione, poiché egli deve aver ricavato la chiave pubblica a partire da quella privata (l'operazione inversa risulta molto più difficile).

Algoritmi noti. Gli algoritmi a chiave simmetrica hanno una coppia di chiavi, per cifratura e de-cifratura, che sono entrambe segrete: DES, AES; si pone il problema di scambiare col destinatario la chiave di de-cifratura, utilizzando un canale sicuro.

Con i sistemi di cifratura a chiave pubblica questo problema non si pone, tuttavia si pone il nuovo problema dell'autenticità delle chiavi pubbliche in circolazione; è stato introdotto il meccanismo dei certificati, da associare alle chiavi pubbliche, per garantire la loro provenienza e affidabilità.

[1] *"Communication Theory of Secrecy Systems"* (1949), Bell System Technical Journal

Numeri interi grandi. Lavoreremo prevalentemente con numeri interi (positivi e negativi) di elevato ordine di grandezza; ecco un esempio per effettuare un calcolo approssimato.

ESEMPIO A.1. *Calcolare in modo approssimato il valore di* 2^{35}.

✓Sfruttando le proprietà delle potenze e la costante informatica $2^{10} \sim 1000 = 10^3$, possiamo ragionare nel modo seguente:

$$2^{35} = 2^{30} \cdot 2^5 = \left(2^{10}\right)^3 \cdot 32 \simeq \left(10^3\right)^3 \cdot 32 = \boxed{32 \times 10^9}$$

□

Trattare interi grandi è importante nell'ambito degli algoritmi di cifratura: prendendo l'algoritmo a chiave simmetrica DES come esempio, è ragionevole pensare che una chiave di 56 bit non sia sufficientemente sicura; usando le considerazioni fatte nell'Esempio A.1 otteniamo che $2^{56} \simeq 10^{16}$, ed essendo in possesso di una macchina in grado di ottenere una chiave in $1ns$, allora sarebbero necessari 27 mesi per ottenere questa chiave; ovviamente si può ridurre questo tempo aumentando il numero di macchine impiegate.

Al giorno d'oggi sono considerate sicure chiavi a 265 bit ($\sim 10^{77}$): per analizzare in modo esaustivo un simile spazio delle chiavi sarebbero necessari 10^{60} anni, nelle condizioni descritte in precedenza!

Indice analitico

Euclide, algoritmo di, 9
Euclide, algoritmo esteso di, 9

Kerchoffs, principio di, 14

Numeri primi, teorema, 7