Appunti di Sicurezza delle reti

Lorenzo Prosseda, a.a. 2018-2019

Indice

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

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

Teorema dei numeri primi

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 a 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 $p_{k=1} = 6 \pm 1 = \{5, 7\}$, $p_{k=2} = 12 \pm 1 = \{11, 13\}$, $p_{k=3} = 18 \pm 1 = \{17, 19\}$, ...; 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 $\text{MCD}(m, n)$. Tramite questo algoritmo otteniamo il risultato desiderato, senza passare per la scomposizione in fattori primi di m e n; definendo la seguente successione:

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

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

$$\text{MCD}(482, 1180) \to 1180 = 2 \cdot 482 + 216$$

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

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

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: $\text{MCD}(482, 1180) = 2$ □

1.3.2. Algoritmo in forma simbolica. Presi $a, b \in \mathbb{Z}$ tali che $a < b$, otteniamo $\text{MCD}(a, b) = d$ 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 $d = \text{MCD}(a, b)$, 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

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

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.

1.4. Congruenza in modulo

1.4.1. Definizione e proprietà. Nelle sezioni precedenti abbiamo introdotto il simbolo di congruenza '$\equiv$', che usato nel modo seguente implica che

$$a \equiv b \,(\text{mod}\, n) \implies a \bmod n = b \bmod n \implies a - b = k \cdot n$$

ovvero il fatto che a sia congruente in modulo n a b implica il fatto che a e b siano multipli; possiamo infatti dedurlo osservando l'ultima implicazione, riscritta come

$$a = b + k \cdot n$$

Per la congruenza in modulo valgono alcune proprietà simili a quelle dell'uguaglianza:

- $a \equiv 0 \,(\text{mod}\, n) \iff n \backslash a$
- $\forall a \in \mathbb{Z} : a \equiv a \,(\text{mod}\, n)$
- $a \equiv b \,(\text{mod}\, n) \iff b \equiv a \,(\text{mod}\, n)$
- $a \equiv b \,(\text{mod}\, n) \wedge b \equiv c \,(\text{mod}\, n) \implies a \equiv c \,(\text{mod}\, n)$

1.4.2. Insieme dei residui. Fin'ora abbiamo lavorato all'interno dell'insieme dei numeri interi $\mathbb{Z}$; introduciamo l'insieme dei residui modulo n, indicato come $\mathbb{Z}_n$, che contiene l'insieme dei valori da 0 a $n-1$ e al suo interno sono definite ciclicamente le operazioni di addizione, sottrazione e moltiplicazione; osserviamo un esempio sull'addizione.

ESEMPIO 1.3. *Determinare i risultati delle seguenti operazioni, all'interno dell'insieme* $\mathbb{Z}_{10}$*:* $4+5,\ 5+5,\ 2-3,\ 3\cdot 4$.

✓Dalla definizione sappiamo che $\mathbb{Z}_{10} = \{0,1,2,3,4,5,6,7,8,9\}$, allora le operazioni richieste valgono:

$$4+5=9,\ 5+5=0,\ 2-3=9,\ 3\cdot 4=2$$

□

La divisione in un insieme dei residui non è definita in modo banale come le altre tre operazioni; introdurremo in seguito questa operazione.

1.4.3. Insieme ridotto dei residui. Dato un insieme dei residui $\mathbb{Z}_n$, possiamo definire il suo insieme ridotto $\mathbb{Z}_n^*$, che contiene gli elementi di $\mathbb{Z}_n$ che sono coprimi rispetto a n, ovvero

$$\forall z \in \mathbb{Z}_n^* \ :\ z \in \mathbb{Z}_n \wedge z \perp n$$

Procedendo con l'Esempio 1.3, l'insieme ridotto dei residui modulo 10 avrà i seguenti elementi al suo interno:

$$\mathbb{Z}_{10}^* = \{1,3,7,9\}$$

Notiamo che lo zero è sempre escluso dall'insieme ridotto, perché non è coprimo rispetto ad alcun intero; la cardinalità di $\mathbb{Z}_n^*$ è determinata da una funzione di n chiamata *toziente* o φ *di Eulero*; in particolare vale $\varphi(10) = 4$ (vedremo in seguito come calcolare questa funzione).

1.4.4. Equazioni congruenziali. Una equazione congruenziale è una relazione definita su un'insieme di residui; nel seguito vedremo delle proprietà che ci permetteranno di operare con queste relazioni.

PROPOSIZIONE 1.1. *Prendiamo un intero* $n \neq 0$ *e quattro interi* a,b,c,d *tali che* $a \equiv b \pmod n$ *e* $c \equiv d \pmod n$*; allora si verifica che*

$$a+c \equiv b+d \pmod n,\ a-c \equiv b-d \pmod n,\ a\cdot c \equiv b\cdot d \pmod n$$

Possiamo usare queste proprietà per risolvere la seguente equazione congruenziale.

ESEMPIO 1.4. *Risolvere rispetto a* x *l'equazione* $x+7 \equiv 3 \pmod{17}$.

✓Dato che 7 è congruente a 3 in modulo 17, possiamo sfruttare la seconda proprietà, per cui

$$x+7\mathbf{-7} \equiv 3\mathbf{-7} \pmod{17}$$
$$x \equiv -4 \pmod{17}$$

Osservando che $17-4=13$, otteniamo infine (è equivalente a scrivere la congruenza con -4)

$$x \equiv 13 \pmod{17}$$

□

Ora introduciamo il concetto di divisione nell'insieme dei residui, con una seconda proposizione.

PROPOSIZIONE 1.2. *Prendiamo un intero* $n \neq 0$ *e tre interi* a,b,c *con* $a \perp n$ *e vale* $a\cdot b \equiv a\cdot c \pmod n$*; allora possiamo dire che* $a\cdot b \equiv a\cdot c \pmod n \implies b \equiv c \pmod n$*, tramite la moltiplicazione da ambo i lati per l'inverso* a^{-1}

Possiamo risolvere una forma di equazione congruenziale più elaborata.

ESEMPIO 1.5. *Risolvere rispetto a x l'equazione* $2x + 7 \equiv 3 \pmod{17}$.

✓Come nel caso precedente, possiamo sottrarre 7 da entrambi i lati, dato che $7 \equiv 3 \pmod{17}$, ottenendo

$$2x \equiv -4 \pmod{17}$$

Ora sfruttando il fatto che $2 \perp 17$ la precedente diventa

$$x \equiv -2 \pmod{17} = x \equiv 15 \pmod{17}$$

□

ESEMPIO 1.6. *Risolvere rispetto a x l'equazione* $5x + 6 \equiv 13 \pmod{11}$.

✓Osservando che $6 \equiv 13 \pmod{11}$, applichiamo la proprietà della divisione e otteniamo

$$5x \equiv 7 \pmod{11}$$

Se ora usiamo la proprietà della divisione (vale $5 \perp 11$) dovremo scrivere

$$x \equiv \frac{\hat{7}}{5} \pmod{11}$$

dove la frazione $\hat{7}/5$ in realtà non esiste nell'insieme $\mathbb{Z}_{11}$; tuttavia sappiamo che $5x$ è congruente modulo 11 a 7, ma anche a (per la ciclicità dell'insieme dei residui) $7 + 11$, $7 + 22$, $\dots$, $7 + k \cdot 11$; il primo numero che sia un multiplo di 5 si ha per 3; abbiamo ottenuto che $[7 = 40] \ni \mathbb{Z}_{11}$, quindi possiamo scrivere

$$5x \equiv 40 \pmod{11}$$
$$x \equiv 8 \pmod{11}$$

Possiamo in un certo senso affermare che 8 si comporta come $7/5$ in $\mathbb{Z}_{11}$.

Esiste una seconda strada per risolvere questo esercizio; sapendo che 5 e 11 sono primi relativi, usiamo il prodotto invece della divisione:

$$5x \cdot \mathbf{5^{-1}} \equiv 7 \cdot \mathbf{5^{-1}} \pmod{11}$$

L'inverso di un numero n è quel numero m tale che $n \cdot m = 1$; dunque cerchiamo l'elemento di $\mathbb{Z}_{11}$ che moltiplicato per 5 risulta 1; troviamo $m \cdot 5 = 1|_{\mathbb{Z}_{11}} \implies m = 9$, da cui segue che

$$x \equiv 7 \cdot 9 \pmod{11}$$

e dato che $7 \cdot 9 = 63 = 5 \cdot 11 + 8$ la precedente si scrive come

$$x \equiv 8 \pmod{11}$$

Abbiamo ottenuto il medesimo risultato a cui siamo arrivati attraverso il primo procedimento. □

OSSERVAZIONE 1.1. Presi $a \in \mathbb{Z}_n$ e $b \notin \mathbb{Z}_n^*$, non è definita la divisione $a \div b$ (deve infatti valere $b \perp n \implies b \in \mathbb{Z}_n^*$).

OSSERVAZIONE 1.2. Se prendiamo l'equazione $a \cdot x \equiv b \pmod{n}$, essa ammetterà soluzione se vale $a \perp n$; in tal caso $\exists a^{-1} \in \mathbb{Z}_n^*$ e la soluzione sarà $x \equiv b \cdot a^{-1} \pmod{n}$.

Cosa possiamo concludere nel caso in cui, data l'equazione $E : a \cdot x \equiv b \pmod{n}$, non sia vero che $a \perp n$, cioè nel caso in cui $\text{MCD}(a, n) = d > 1$? In tal caso, l'equazione può non ammettere soluzione o ammetterne d.

Se $d \backslash b$ allora dividiamo per d tutte le quantità costanti, ottenendo

$$\overline{E} : \frac{a}{d}x \equiv \frac{b}{d} \left(\text{mod } \frac{n}{d}\right)$$

questa equazione ha ora una soluzione, perché $\text{MCD}(a,\, n) = d \implies \text{MCD}\left(\frac{a}{d},\, \frac{n}{d}\right) = \frac{d}{d} = 1$ e ricadiamo nel caso dell'osservazione precedente.

Otteniamo la soluzione dell'equazione $\overline{E}$ e la chiamiamo x_0, la quale sarà in modulo n/d ($x_0 \in \mathbb{Z}_{n/d}$); dato che l'equazione di partenza E ha soluzioni in $\mathbb{Z}_n$, esse saranno i termini della successione

$$\{x\} = x_0,\, x_0 + \frac{n}{d},\, x_0 + 2\frac{n}{d},\, \dots\, ,\, x_0 + (d-1)\,\frac{n}{d}$$

Osserviamo l'esempio di un'equazione con queste caratteristiche.

ESEMPIO 1.7. *Risolvere rispetto a x l'equazione* $12x \equiv 21 \pmod{39}$.

✓Osserviamo subito che $\text{MCD}(12,\, 39) = 3$, dunque l'equazione avrà 0 o 3 soluzioni; dividiamo le costanti per 3 e otteniamo

$$4x \equiv 7 \pmod{13}$$

da cui otteniamo $x \equiv \frac{\hat{7}}{4} \pmod{13}$ e trovando $4 \backslash (7 + 1 \cdot 13) = 20$ la precedente equazione fornisce una soluzione $x_0 \equiv 5 \pmod{n}$; adesso usiamo la successione delle soluzioni $\{x\}$ per ottenere tutte le soluzioni dell'equazione di partenza:

$$x = \left\{ x_0 = \boxed{5},\, x_0 + \frac{39}{3} = \boxed{18},\, x_0 + \frac{2 \cdot 39}{3} = \boxed{31} \right\}$$

Possiamo immaginare le radici di E come dei punti su una circonferenza, a distanza d/n da x_0. □

Posto che valga $a \perp n$ (ovvero $\text{MCD}(a,\, n) = 1$), come possiamo calcolare l'inverso a^{-1}? Potremmo tentare tutti i numeri da 1 a $n-1$ conducendo un'analisi esaustiva, tuttavia per n molto grande questo non è pratico; usiamo allora l'algoritmo di Euclide esteso (1.3.4): esso ci garantisce che

$$\text{MCD}(a,\, n) = 1 \implies \exists s, t \in \mathbb{Z}_n \,:\, a \cdot s + n \cdot t = 1$$

segue dalla precedente relazione che $a \cdot s = 1 - n \cdot t$, allora abbiamo $a \cdot s \equiv 1 \pmod{n}$ e risolvendo l'equazione si ottiene

$$s \equiv a^{-1} \pmod{n}$$

ovvero il numero s è proprio l'elemento finale x_{k+1} della sequenza di x dell'algoritmo; esso risulta essere anche il valore cercato dell'inverso a^{-1}.

1.5. Teorema cinese del resto

1.5.1. Applicazione ed enunciato. Mostriamo direttamente l'applicazione del teorema a un caso specifico: prendiamo $x \equiv 25 \pmod{42}$; possiamo esprimere $x = 25 + 6 \cdot (7 \cdot k) \rightleftarrows 25 + 7 \cdot (6 \cdot k)$, da cui ricaviamo

- per $x = 25 + 6 \cdot (7 \cdot k)$: $25 \bmod 6 = 1 \implies x \equiv 1 \pmod{6}$
- per $x = 25 + 7 \cdot (6 \cdot k)$: $25 \bmod 7 = 3 \implies x \equiv 3 \pmod{7}$

Dalle due congruenze più semplici ottenute possiamo scrivere il sistema

$$x \equiv 25 \pmod{42} \implies \begin{cases} x \equiv 1 \pmod{6} \\ x \equiv 3 \pmod{7} \end{cases}$$

Il teorema cinese del resto afferma che:

Teorema cinese del resto

TEOREMA 1.2. *Dati due interi n, m che siano primi relativi (deve valere $n \perp m$), e presi due interi a, b, allora il sistema delle congruenze*

$$\begin{cases} x \equiv a \,(\mathrm{mod}\, n) \\ x \equiv b \,(\mathrm{mod}\, m) \end{cases}$$

è equivalente alla singola congruenza

$$x \equiv c \,(\mathrm{mod}\, n \cdot m)$$

Il teorema afferma che, quanto introdotto all'inizio della sezione, può valere nel senso opposto in alcune condizioni specifiche.Osserviamo l'applicazione del teorema in un esempio.

ESEMPIO 1.8. *Sia dato il seguente insieme di congruenze:*

$$\begin{cases} x \equiv 3 \,(\mathrm{mod}\, 7) \\ x \equiv 5 \,(\mathrm{mod}\, 15) \end{cases}$$

Scrivere una forma equivalente, con una singola congruenza.

✓Dato che $7 \perp 15$, per il Teorema 1.2 possiamo affermare che esisterà una singola congruenza equivalente alle due della consegna; essa sarà scritta nella forma $x \equiv c \,(\mathrm{mod}\, 105)$. Per ottenere c possiamo provare i numeri multipli di 5, i quali in modulo 7 diano 3 come risultato; si ottiene facilmente $c = 80$ (possiamo scomporlo come $80 = 11 \cdot 7 + 3$).

Nel caso di numeri molto grandi, non è possibile scegliere di procedere per tentativi, ed è necessario usare gli strumenti della teoria dei numeri: se sappiamo che $x \equiv a \,(\mathrm{mod}\, n)$ e anche $x \equiv b \,(\mathrm{mod}\, m)$, allora possiamo scrivere

$$x = b + \overline{k} \cdot m \equiv a \,(\mathrm{mod}\, n)$$

risolvendo questa relazione si ottiene $a - b \equiv \overline{k} \cdot m \,(\mathrm{mod}\, n)$ da cui, rispetto a $\overline{k}$ otteniamo

$$\overline{k} \equiv (a - b) \cdot m^{-1} \,(\mathrm{mod}\, n)$$

Il valore $\overline{k}$ corrisponde al c del Teorema 1.2, esso infatti è congruente sia ad a in modulo n, sia a b in modulo m. □

1.5.2. Estensione. Il Teorema 1.2 ammette un'estensione a un caso generale, con N congruenze.

Teorema cinese del resto esteso

COROLLARIO 1.1. *Prendiamo N numeri interi, indicati come $m_i : i \in [1, N]$, tali che a coppie siano tutti coprimi ($\forall i, j \in [1, N] \wedge i \neq j : m_i \perp m_j$). Se abbiamo il seguente sistema di congruenze*

$$\begin{cases} x \equiv a_1 \,(mod\, m_1) \\ x \equiv a_2 \,(mod\, m_2) \\ \\ x \equiv a_N \,(mod\, m_N) \end{cases}$$

è equivalente alla singola congruenza

$$x \equiv C \left(mod \prod_{i=1}^{N} m_i \right)$$

Tale congruenza si costruisce a partire dalle seguenti serie:

$$M = \prod_{i=1}^{N} m_i; \quad Z_i = \frac{M}{m_i}; \quad Y_i = Z_i^{-1} \bmod m_i$$

Otteniamo infine la seguente soluzione, equivalente a una singola congruenza:

$$X = \sum_{i=1}^{N} a_i Y_i Z_i \bmod M \tag{1.5.1}$$

1.6. Square & multiply

Tramite l'algoritmo chiamato square and multiply, è possibile calcolare il modulo di un numero che abbia molte cifre; mostriamo un esempio per illustrare l'idea dietro a questo algoritmo, usando numeri "piccoli".

ESEMPIO 1.9. *Si calcoli il risultato dell'operazione* $7^{11} \bmod 26$.

✓Eseguiamo i seguenti passi per cercare il valore desiderato:

LSB indica il least significant bit (bit meno significativo) e si riferisce al primo bit da destra

(1) Esprimiamo 11 (l'esponente) in forma binaria: $11_{10} = 1011_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$;
(2) Sostituire 11 con la sua rappresentazione binaria: $7^{11} = 7^{(8+2+1)} = 7^8 \cdot 7^2 \cdot 7$;
(3) Scriviamo i bit di 11 in colonna, al contrario (a partire dal LSB): $\begin{bmatrix}1 & 1 & 0 & 1\end{bmatrix}$;
(4) Scriviamo accanto a ciascun bit la potenza di 7 corrispondente, ottenuta dalla scomposizione al punto (1);
(5) Scriviamo accanto alle potenze 7^i il risultato dell'operazione $7^i \bmod 26$;
(6) Componiamo il risultato prendendo il prodotto dei valori scritti al punto (5) in corrispondenza di un 1, scritto al punto (3)

Otterremo infine:

$$\begin{array}{ccc} \mathbf{1} & 7^1 & \boxed{7} \\ \mathbf{1} & 7^2 & \boxed{23} \\ 0 & 7^4 & 9 \\ \mathbf{1} & 7^8 & \boxed{3} \end{array} \implies (7 \cdot 23 \cdot 3) \bmod 26 = 15$$

Il vantaggio di questo metodo è il fatto che si basa solo sull'operazione di elevamento al quadrato di numeri piccoli (relativamente al modulo, nell'esercizio siamo in $\mathbb{Z}_{26}$). In totale, saranno necessari (per questo caso dell'esercizio) $2 \cdot \log_2(11) \simeq 4$. □

1.7. Teorema piccolo di Fermat

Teorema piccolo di Fermat

TEOREMA 1.3. *Sia p un intero primo, e a un intero tale che p non divida a, allora è vero che*

$$a^{p-1} \equiv 1 \pmod{p} \tag{1.7.1}$$

Non vale l'implicazione inversa, infatti prendendo a ed n due interi qualsiasi, e osservando che $a^{n-1} \equiv 1 \pmod{n}$, non possiamo dedurre che n sia primo.

Piuttosto possiamo usare il Teorema 1.3 per provare che un numero sia composto (non primo); tuttavia i numeri composti che danno resto 1 non ostante non siano primi sono pochi, e sono definiti *pseudo-primi*. Questi numeri sono sempre più rarefatti all'aumentare dell'ordine di grandezza.

Non ostante la scelta di una base a differente possa "smascherare" un numero pseudo-primo, esiste una categoria di interi chiamati *pseudo-primi assoluti*, i quali forniscono resto 1 con qualunque base scelta nel test del teorema piccolo di Fermat.

Osservazione 1.3. Abbiamo concluso dal Teorema 1.3 che, se non è vero $p \backslash a$, allora vale $a^{p-1} \equiv 1 \pmod{p}$; possiamo scrivere questa congruenza in modo equivalente come $a \cdot a^{p-2} \equiv 1 \pmod{p}$, che può essere risolta con la regola della divisione nel modo seguente:

$$a^{p-2} \equiv a^{-1} \pmod{p} \tag{1.7.2}$$

Abbiamo appena ottenuto un nuovo metodo per calcolare l'inverso di un intero a.

Poniamoci nel caso generale, in cui abbiamo un intero qualunque a e un intero composto n: la condizione del teorema diventa che $a \perp n$; in queste condizioni possiamo enunciare il seguente Teorema 1.4.

Teorema 1.4. *Sia a un intero qualunque e n un intero composto, tali che $a \perp n$, allora è vero che*

Teorema di Eulero

$$a^{\varphi(n)} \equiv 1 \,(mod\, n) \tag{1.7.3}$$

La funzione *toziente* $\varphi(p)$ con p numero primo, è il numero di primi relativi rispetto a p, in questo caso $p-1$ numeri, tutti i predecessori di p, il quale essendo primo ha come fattore comune con essi solo 1; vale dunque $\varphi(p) = p-1$ per p numero primo.

In generale il toziente di un intero n qualunque si può calcolare in uno dei seguenti modi:

Toziente

- $\varphi(n) = n \cdot \prod_{i \in [1,n-1],\, p_i \backslash n} \left(1 - \frac{1}{p_i}\right)$
- $n = \prod_{i=1}^{m} p_i^{r_i} \implies \varphi(n) = \prod_{i=1}^{m} \left(p_i^{r_i} - p_i^{r_i - 1}\right)$

Vediamo un esempio in cui calcoliamo il toziente di due numeri adoperando entrambi i procedimenti:

Esempio 1.10. *Calcolare il toziente di 1000.*

✓Adoperando il primo metodo, scriviamo la scomposizione di 1000 nei suoi fattori primi: $1000 = 2^3 \cdot 5^3$; scriviamo dunque la produttoria usando i due fattori $p_i = \{2, 5\}$ senza esponente

$$1000 \cdot \left(1 - \frac{1}{2}\right) \cdot \left(1 - \frac{1}{5}\right) = \boxed{400} = \varphi(1000)$$

Se adoperiamo il secondo metodo, scriviamo direttamente la produttoria a partire dai fattori primi 2^3 e 5^3:

$$\left(2^3 - 2^2\right) \cdot \left(5^3 - 5^2\right) = \boxed{400} = \varphi(1000)$$

Abbiamo ottenuto lo stesso toziente, mostrando che entrambi i procedimenti si equivalgono. □

Osservazione 1.4. Se vogliamo calcolare l'inverso di un intero a modulo n non primo, posto che $a \perp n$ possiamo scrivere la congruenza

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

Questa relazione ci permette di calcolare un inverso utilizzando il modulo di un elevamento a potenza (evitando la strada di Euclide esteso (1.3.4)), per esempio tramite square and multiply eseguito su un calcolatore.

Mettendo assieme quanto illustrato in questa sezione, possiamo rispondere alla richiesta mostrata nel seguente esempio.

ESEMPIO 1.11. *Calcolare le ultime tre cifre di* 7^{803}.

✓Le ultime tre cifre di un numero possono essere ottenute usando il resto della sua divisione per 10^3; allora dobbiamo trovare il risultato di $7^{803} \bmod 1000$.

Per il teorema di Eulero (1.7.3) sappiamo che $7^{\varphi(1000)} \equiv 1 \pmod{1000}$ e dal precedente esempio abbiamo già calcolato $\varphi(1000) = 400$, per cui possiamo scomporre 7^{803} nel modo seguente:

$$\left(7^{400} \cdot 7^{400} \cdot 7^3\right) \bmod 1000 = (1 \cdot 1 \cdot 343) \bmod 1000 = \boxed{343}$$

□

1.8. Principio fondamentale

Se abbiamo gli interi a, b, n, x, y e $n > 0$, tali che $a \perp n$, allora se $x \equiv y \pmod{\varphi(n)}$ vale

$$a^x \equiv b^y \pmod{n} \tag{1.8.1}$$

tuttavia non vale l'implicazione inversa.

Concludiamo che, in una congruenza modulo n, in presenza di elevamenti a potenza gli esponenti della stessa base lavorano in modulo di $\varphi(n)$.

Questo principio permette di risolvere rapidamente la richiesta dell'Esercizio 1.11: dato che $803 \equiv 3 \pmod{\varphi(1000) = 400}$ otteniamo direttamente $7^{803} \equiv 7^3 \pmod{1000}$; abbiamo usato (1.8.1) in cui $x = 803$ e $y = 3$.

1.9. Radici in aritmetica modulare

1.9.1. Radici primitive. Introduciamo il concetto di gruppo algebrico:

Gruppo algebrico

Chiamiamo gruppo *una struttura formata da un insieme e da un'operazione binaria definita su di esso, la quale soddisfi gli assiomi di associatività, esistenza dell'elemento neutro ed esistenza dell'elemento inverso*

Ad esempio, se prendiamo l'insieme dei numeri interi $\mathbb{Z}$ e l'operazione di somma $+$, possiamo formare il gruppo $(\mathbb{Z}, +)$ poiché la somma è associativa, l'elemento neutro è lo zero e l'inverso di qualunque elemento è sempre definito; anche prendendo $\mathbb{Z}_p^*$ (insieme dei residui interi modulo p, escluso lo 0) e l'operazione di prodotto, otteniamo di nuovo un gruppo $\left(\mathbb{Z}_p^*, \cdot\right)$. Non si ottiene un gruppo rispetto al prodotto per $\mathbb{Z}_p$ (residui modulo p, da 0 a $p-1$), poiché lo 0 non ha un inverso definito.

Se prendiamo un elemento $a \in \mathbb{Z}_p^*$ allora vale $a^{p-1} \equiv 1 \pmod{p}$ per il Teorema 1.3; chiamiamo inoltre ordine dell'elemento a (e lo indichiamo tramite $\text{ORD}(a)$) l'intero $n > 0$ più piccolo tale che

$$a^n \equiv 1 \pmod{p} \tag{1.9.1}$$

Il teorema piccolo di Fermat (1.7.1) ci assicura che vale sempre $n \leq p - 1$, ma non è detto che sia proprio $n = p - 1$. Chiamiamo quindi α l'elemento primitivo $a \in \mathbb{Z}_p^*$ se e solo se l'ordine di a è 1, ovvero non esiste nessun altro intero n per cui $a^n \equiv 1 \pmod{p}$; in altri termini vale

$$a^n \equiv 1 \pmod{p} \iff n = 1 \tag{1.9.2}$$

Se l'ordine di a fosse proprio $p - 1$, allora preso α^i l'elemento primitivo, presi tutti gli interi $i \in [1, p-1]$, otteniamo dalla successione degli α^i tutti e soli gli elementi dell'insieme $\mathbb{Z}_p^*$ (in ordine anche differente); chiameremo l'elemento α la *radice primitiva*.

Radice primitiva

Chiamiamo α una radice primitiva per p se vale (1.9.2) e inoltre l'ordine di α è pari a $n = p - 1$.

ESEMPIO 1.12. *Trovare se $\alpha = 3$ sia una radice primitiva di $p = 7$.*

✓Per trovare se α sia effettivamente una radice primitiva di p, calcoliamo gli elementi:

$$\left\{\forall i \in [0,\, p-1] \,:\, \beta \equiv \alpha^i \,(\mathrm{mod}\, p)\right\}$$

Otteniamo quindi:

$$\begin{aligned} 3^0 &\equiv 1 \,(\mathrm{mod}\, 7)\\ 3^1 &\equiv 3 \,(\mathrm{mod}\, 7)\\ 3^2 &\equiv 2 \,(\mathrm{mod}\, 7)\\ 3^3 &\equiv 6 \,(\mathrm{mod}\, 7)\\ 3^4 &\equiv 4 \,(\mathrm{mod}\, 7)\\ 3^5 &\equiv 5 \,(\mathrm{mod}\, 7)\\ &\vdots \end{aligned}$$

Abbiamo riottenuto tutti e soli gli elementi da 1 a $p - 1 = 6$ (i membri di $\mathbb{Z}_7^*$), con periodo $p - 1 = 6$. Si verifica dunque la condizione che rende un elemento primitivo α una radice per p; la radice primitiva è anche detta elemento generatore: infatti, tramite essa è possibile ottenere tutti i residui in modulo p di un certo insieme $\mathbb{Z}_p^*$. □

ESEMPIO 1.13. *Trovare se $\alpha = 2$ sia una radice primitiva di $p = 7$.*

✓Calcolando di nuovo gli α^i con $i \in [0,\, p-1]$ otteniamo

$$\begin{aligned} 2^0 &\equiv 1 \,(\mathrm{mod}\, 7)\\ 2^1 &\equiv 2 \,(\mathrm{mod}\, 7)\\ 2^2 &\equiv 4 \,(\mathrm{mod}\, 7)\\ 2^3 &\equiv 1 \,(\mathrm{mod}\, 7)\\ &\vdots \end{aligned}$$

In questo caso abbiamo ottenuto periodo 3; inoltre l'ordine di α è $n = 3 \neq p-1 = 6$, quindi $\alpha = 2$ non è radice primitiva di $p = 7$. □

OSSERVAZIONE 1.5. Se prendiamo l'elemento primitivo $a \in \mathbb{Z}_p^*$, esso avrà al massimo ordine $p - 1$; tuttavia potrà presentare anche ordini pari ai sottomultipli di $p - 1$, ma non ordini differenti da essi.

OSSERVAZIONE 1.6. Se α è un elemento primitivo di $\mathbb{Z}_p^*$ e vale $\beta \equiv \alpha^i \,(\mathrm{mod}\, p)$ con $1 \leq i \leq p - 1$, questi β sono tutti e soli gli elementi di $\mathbb{Z}_p^*$. Se ora prendiamo un elemento primitivo di $\mathbb{Z}_p^*$ espresso come β (tramite la precedente congruenza), il suo ordine sarà dipendente dall'esponente i usato per ottenerlo a partire da α,e dovrà valere

$$\mathrm{ORD}\,(\beta) = \frac{p-1}{\mathrm{MCD}\,(p-1,\, i)} \tag{1.9.3}$$

dove i è l'esponente a cui elevare α per ottenere β. Questo si verifica perché i βsono potenze di α, ed essendo $\mathrm{ORD}\,(\alpha) = p - 1$ (α è radice primitiva) necessariamente l'ordine di β sarà un sottomultiplo di $p - 1$, determinato da (1.9.3)

Gli elementi primitivi di $\mathbb{Z}_p^*$ sono in numero $\varphi\,(p-1)$; infatti saranno tutti gli elementi di $\mathbb{Z}_p^*$. coprimi rispetto a $p - 1$, per cui si verifica la condizione (1.9.3). Notiamo infatti che β è una radice primitiva di p (vale $\mathrm{ORD}\,(\beta) = p - 1$) quando si ha$\mathrm{MCD}\,(p-1,\, i) = 1$; concludiamo che il numero di elementi primitivi β di un insieme $\mathbb{Z}_p^*$ è il numero di elementi coprimi rispetto a $p - 1$ (valore ottenuto dalla funzione toziente di $p - 1$).

1.9.2. Test di primitività. Prendiamo un $\alpha \in \mathbb{Z}_p^*$ con p qualunque, e cerchiamo se α sia primitivo o meno rispetto a p; usiamo il seguente test di primitività: scomporre $p-1$ nei suoi fattori primi, in modo da avere una produttoria di quozienti elevati a un certo esponente $p-1 = \prod_i q_i^{r_i}$; diremo che α è primitivo rispetto a p se e solo se vale

Test primitività

$$\forall i \,:\, \alpha^{(p-1)/q_i} \not\equiv 1 \,(\mathrm{mod}\, p) \tag{1.9.4}$$

Si noti che p è primo (per definizione), di conseguenza $p-1$ è sempre pari.

Esempio 1.14. *Ottenere se $\alpha = 2$ sia radice primitiva per $p = 19$.*

✓Usando il test (1.9.4), scomponiamo $19 - 1 = 18$ nei suoi fattori primi:

$$18 = 2 \cdot 3^2$$

Ora testiamo la congruenza per tutti i fattori primi (presi senza esponente):

$$2^{18/3} = 2^6 = 64 \equiv 7 \,(\mathrm{mod}\, 19)$$
$$2^{18/2} = 2^9 = 512 \equiv 18 \,(\mathrm{mod}\, 19)$$

Per nessun fattore primo si ha congruenza a 1 modulo 19, dunque $\alpha = 2$ è davvero una radice primitiva per $p = 19$.

Osserviamo infine che gli elementi primitivi di $\mathbb{Z}_{19}$ sono $\varphi(18) = 6$, distribuiti all'interno dell'insieme in modo non predicibile. □

1.9.3. Radici quadrate. Poniamoci nel caso di radici quadrate in modulo a un intero p primo; inoltre, consideriamo per tale intero solamente la metà dei numeri primi (si ricordi la separazione dei primi in classi di congruenza (1.2.1)):

$$p \equiv 3 \,(\mathrm{mod}\, 4) \tag{1.9.5}$$

Risolviamo, sotto queste condizioni, l'equazione $x^2 \equiv a \,(\mathrm{mod}\, p)$, ovvero cerchiamo $\sqrt{a} \in \mathbb{Z}_p$; si dimostra che, se esiste $\sqrt{a}$, allora vale:

Radice quadrata

$$\pm\, a^{(p+1)/4} = \sqrt{a} \;\vee\; \pm a^{(p+1)/4} = \sqrt{-a} \tag{1.9.6}$$

dove abbiamo l'unione di due condizioni che sono esclusive (se non esiste la radice di a allora esisterà quella di $-a$, e vale quanto indicato in (1.9.6), e vice versa).

Osservazione 1.7. Se avessimo $p \not\equiv 3 \,(\mathrm{mod}\, 4)$ ma $p \equiv 1 \,(\mathrm{mod}\, 4)$ (p appartenente all'altra classe di congruenze), allora nel caso in cui non esista una delle due radici ($\sqrt{a}$ o $\sqrt{-a}$) non esiste nemmeno l'altra.

Esempio 1.15. *Si calcoli, se esiste, la radice di 5 in modulo 11.*

✓Cercare $\sqrt{5} \,\mathrm{mod}\, 11$ equivale a risolvere l'equazione

$$x^2 \equiv 5 \,(\mathrm{mod}\, 11)$$

Dato che per 11 vale (1.9.5), possiamo usare la formula (1.9.6):

$$\pm 5^{(11+1)/4} = \pm 125 \,\mathrm{mod}\, 11 = \pm 4$$

Per determinare se in modulo 11 i numeri ± 4 siano la radice di 5 o di -5, effettuiamo l'elevamento a potenza della radice cercata:

$$4^2 \,\mathrm{mod}\, 11 = 5, \quad -4^2 = 7^2 \,\mathrm{mod}\, 11 = 5$$

Abbiamo ottenuto che la radice quadrata di 5 modulo 11 è ± 4. □

Esempio 1.16. *Si calcoli, se esiste, la radice di 2 in modulo 11.*

✓Come prima, impostiamo così l'equazione da risolvere:

$$x^2 \equiv 2 \pmod{11}$$

Possiamo usare (1.9.6), dato che per 2 vale (1.9.5):

$$\pm 2^{(11+1)/4} = \pm 8 \bmod 11 = \pm 8$$

Infine effettuiamo l'elevamento a potenza della radice per ottenere il segno:

$$8^2 \bmod 11 = 9 \bmod 11 = -2$$

Abbiamo ottenuto che in modulo 11 non esiste la radice quadrata di 2; la radice di -2 però esiste, e vale ± 8. □

1.9.4. Test per segno della radice. Prendiamo un primo p dispari, un $a \neq 0$ in modulo p, allora

$$a^{(p-1)/2} \equiv \pm 1 \pmod{p} \tag{1.9.7}$$

Distinguiamo i due casi:

- se il segno in (1.9.7) è $+$, allora esiste $\sqrt{a}$ in modulo p;
- se il segno in (1.9.7) è $-$, allora esiste $\sqrt{-a}$ in modulo p.

Ricordando l'Osservazione 1.7, si nota che il test appena mostrato permette di affermare se la radice non sia definita, nel caso in cui $p \equiv 1 \pmod 4$ e in cui il segno in (1.9.7) sia $-$.

1.9.5. Radice quadrata modulo composto. Esaminiamo il problema $x^2 \equiv a \pmod n$ dove n sia un numero composto; dato che n sarà scomponibile in fattori primi, possiamo scomporre la congruenza esaminata in due congruenze più semplici in modulo p e q, sapendo che $n = p \cdot q$. Le due soluzioni ottenute si combinano infine tramite il Teorema 1.2.

Questo problema ha la sua difficoltà nella fattorizzazione di n, al punto che la complessità computazionale del calcolo della radice modulo n equivale a quella della sua scomposizione in fattori primi.

Esempio 1.17. *Si calcoli, se esiste, la radice di 71 in modulo 77.*

✓Notiamo subito che $77 = 7 \cdot 11$; per il Teorema 1.2 possiamo scomporre la congruenza in esame nelle due più semplici:

$$x^2 \equiv 71 \pmod{77} \implies \begin{cases} x^2 \equiv 1 \pmod 7 \\ x^2 \equiv 5 \pmod{11} \end{cases}$$

Nella precedente abbiamo già sostituito $71 \bmod 7 = 1$ e $71 \bmod 11 = 5$; possiamo risolvere le due congruenze in modo indipendente:

(1) la radice quadrata di 1 è comunque ± 1, modulo 7;
(2) la radice quadrata di 5 modulo 11 si può ottenere da (1.9.6) e risulta $x \equiv \pm 5^{(11+1)/4} \pmod{11} \implies x \equiv \pm 4 \pmod{11}$.

Abbiamo ottenuto due radici da ciascuna congruenza, che possono essere combinate in $2^2 = 4$ modi possibili, per costruire la soluzione tramite il Teorema 1.2.

- $\{x \equiv 1 \pmod 7 \,;\; x \equiv 4 \pmod{11}\}$
  Risolviamo la combinazione delle due congruenze analizzate:

$$\begin{gathered} x = 1 + k \cdot 7 \equiv 4 \pmod{11} \\ 7 \cdot k \equiv 3 \pmod{11} \\ k \equiv 3 \cdot 7^{-1} \pmod{11} \end{gathered}$$

Per trovare l'inverso utilizziamo (1.7.2), da cui $7^{-1} \bmod 11 = 7^9 \bmod 11 = 8$, da cui segue che $k = 24 \bmod 11 = 2$; sostituendo nella prima si ottiene

$$x = 14 + 1 \equiv 4 \,(\mathrm{mod}\, 11) \implies \boxed{x \equiv 15 \,(\mathrm{mod}\, 77)}$$

dal Teorema 1.2.

- $\{x \equiv 1 \,(\mathrm{mod}\, 7)\,;\ x \equiv -4 \,(\mathrm{mod}\, 11)\}$

$$x \equiv -15 \,(\mathrm{mod}\, 77)$$

- $\{x \equiv -1 \,(\mathrm{mod}\, 7)\,;\ x \equiv 4 \,(\mathrm{mod}\, 11)\}$

$$x \equiv 29 \,(\mathrm{mod}\, 77)$$

- $\{x \equiv -1 \,(\mathrm{mod}\, 7)\,;\ x \equiv -4 \,(\mathrm{mod}\, 11)\}$

$$x \equiv -29 \,(\mathrm{mod}\, 77)$$

Non è detto che le radici siano sempre 4, in questo caso esistevano tutte le radici per ciascuna congruenza in cui è stata scomposta quella iniziale. □

1.9.6. Residui quadratici. Abbiamo chiamato gli elementi di $\mathbb{Z}_p^*$ residui modulo p; i residui quadratici sono residui che corrispondono anche al quadrato di qualche elemento dello stesso insieme $\mathbb{Z}_p^*$.
Chiamiamo $a_q \in \mathbb{Z}_p^*$i residui quadratici per cui valga

$$a_q \equiv (\pm b)^2 \,(\mathrm{mod}\, p)$$

con $b \in \mathbb{Z}_p^*$ un elemento dell'insieme a cui appartiene anche a_q.

Osserviamo che $p \in \left\{1, 2, \ldots, \frac{p-1}{2}\right\}$, poiché i residui a partire da $\frac{p-1}{2} + 1$ sono i residui negativi dei precedenti dal minore al maggiore $(-1 = p - 1)$.

Esempio 1.18. *Trovare i residui quadratici dell'insieme* $\mathbb{Z}_{11}^*$.

✓Otteniamo subito che nell'insieme analizzato ci sono $\frac{p-1}{2} = \frac{11-1}{2} = 5$ residui quadratici; possiamo ottenerli tramite il test (1.9.7):

$$a^{(p-1)/2} \not\equiv +1 \,(\mathrm{mod}\, p) \implies a \textbf{ non } \text{è residuo quadratico!}$$

Effettuiamo questo test per ciascun elemento di $\mathbb{Z}_{11}^*$:

$$\begin{array}{ll} 1^5 \equiv \boxed{1} \,(\mathrm{mod}\, 11) & 6^5 \equiv -1 \,(\mathrm{mod}\, 11) \\ 2^5 \equiv -1 \,(\mathrm{mod}\, 11) & 7^5 \equiv -1 \,(\mathrm{mod}\, 11) \\ 3^5 \equiv \boxed{1} \,(\mathrm{mod}\, 11) & 8^5 \equiv -1 \,(\mathrm{mod}\, 11) \\ 4^5 \equiv \boxed{1} \,(\mathrm{mod}\, 11) & 9^5 \equiv \boxed{1} \,(\mathrm{mod}\, 11) \\ 5^5 \equiv \boxed{1} \,(\mathrm{mod}\, 11) & 10^5 \equiv -1 \,(\mathrm{mod}\, 11) \end{array}$$

Raccogliendo i risultati dei test con la congruenza a $+1$, possiamo scrivere l'insieme dei residui quadratici di $\mathbb{Z}_{11}^*$:

$$a_q = \{1, 3, 4, 5, 9\}$$

□

Osservazione 1.8. Un modo più rapido per ottenere i residui quadratici consiste nel sfruttare il fatto che essi saranno i quadrati dei primi $\frac{p-1}{2}$ elementi dell'insieme dei residui:

$$a_q = \left\{ 1^2, \ldots, \left(\frac{p-1}{2} \right)^2 \right\}$$

Riprendendo la consegna dell'Esempio 1.18, i residui quadratici di $\mathbb{Z}_{11}^*$ si ottengono come:

$$\begin{array}{ll} 1^2 \bmod 11 = 1 & 3^2 \bmod 11 = 9 \quad 5^2 \bmod 11 = 3 \\ 2^2 \bmod 11 = 4 & 4^2 \bmod 11 = 5 \end{array} \implies a_q = \{1, 4, 9, 5, 3\}$$

Otteniamo gli stessi residui quadratici trovati con i test, in ordine diverso.

CAPITOLO 2

Cifrari elementari

2.1. Introduzione

Facendo riferimento alla figura A.0.1, possiamo individuare la caratteristica del sistema di cifratura modellizzato: esso usa la stessa chiave k per la cifratura e la decifratura del messaggio; ammettiamo che essa sia consegnata al destinatario tramite un canale sicuro.

Il testo in chiaro è definito all'interno di tutti i possibili messaggi in chiaro componibili con l'alfabeto a disposizione ($p \in \mathscr{P}$), il messaggio cifrato è anch'esso definito all'interno di un insieme di ogni possibile messaggio cifrato componibile ($c \in \mathscr{C}$), infine la chiave appartiene all'insieme di tutte le possibili chiavi componibili ($k \in \mathscr{K}$).

Ogni chiave, all'interno dello spazio delle chiavi, è definita una regola di cifratura ($\forall k \in \mathscr{K} \ : \ E_k \in \mathscr{E}$) e anche una regola di decifratura ($\forall k \in \mathscr{K} \ : \ D_k \in \mathscr{D}$), entrambe dipendenti da k.

Deve valere che la cifratura sia invertibile con la decifratura, ovvero $\forall k \in \mathscr{K}$, $p \in \mathscr{P}$, $c \in \mathscr{C} \ : \ c = E_k(p) \wedge p = D_k(c)$; in caso contrario la decifratura non sarebbe più possibile (collisioni durante la cifratura di messaggi differenti).

2.2. Cifrari a scorrimento e sostituzione

2.2.1. Cifrario di Cesare. Il cifrario di Cesare utilizza uno scorrimento dell'alfabeto, la chiave stessa è un simbolo dell'alfabeto:

$$\mathscr{P} = \mathscr{C} = \mathscr{K} = \mathbb{Z}_n$$

Si verifica nel caso dell'alfabeto inglese vale $n = 26$; la funzione di cifratura e di decifratura si indicano come:

$$c = E_k(p) = (p + k) \bmod n$$
$$p = D_k(c) = (c - k) \bmod n$$

Possiamo attaccare questo cifrario con una ricerca esaustiva dello spazio delle chiavi, sapendo che $|\mathscr{K}| = n$, oppure possiamo confrontare un testo in chiaro noto con un testo cifrato per ottenere la chiave ($k = p - c$).

Chiamiamo questo tipo di cifrario *mono-alfabetico*, ovvero ad ogni carattere corrisponde un carattere.

2.2.2. Cifrario affine. Utilizza una combinazione lineare applicata al testo in chiaro; per questo cifrario vale:

$$\mathscr{P} = \mathscr{C} = \mathbb{Z}_n,\ \mathscr{K} = k(a,\, b)$$

La funzione di cifratura e di decifratura si indicano come:

$$c = E_k(p) = (ap + b) \bmod n$$
$$p = D_k(c) = (c - b)\, a^{-1} \bmod n$$

Bisogna prendere $a,\ b \in \mathbb{Z}_n$ e per la presenza dell'inversione di a, è necessario che esso sia primo relativo a n ($a \perp n$); se così non fosse, si verificherebbero delle collisioni nella cifratura.

2.2.3. Cifrario a sostituzione. Ogni carattere è sostituito da un'altro, secondo una tabella arbitraria; esso è la generalizzazione dei due precedenti cifrari (la chiave è una permutazione dell'alfabeto), e vale

$$\mathscr{P} = \mathscr{C} = \mathbb{Z}_n,\ |\mathscr{K}| = n!$$

La chiave è la tabella di sostituzione, e le funzioni di cifratura e decifratura sono determinate dalle sostituzioni indicate nella tabella.

Possiamo attaccare questo cifrario (e tutti gli altri cifrari mono-alfabetici) analizzando l'entropia dei singoli simboli dell'alfabeto: nel caso di messaggi cifrati in linguaggio naturale, l'analisi delle frequenze dei simboli può rivelare lettere molto usate.

Altri attacchi di questo tipo includono l'analisi della frequenza dei digrammi e trigrammi (coppie e triple di lettere).

2.2.4. Cifrario di Vigenère. Questo cifrario lavora su blocchi di h caratteri alla volta, e per questo viene detto poli-alfabetico. In questo caso abbiamo:

$$\mathscr{P} = \mathscr{C} = \mathscr{K} = (\mathbb{Z}_n)^h$$

La chiave, il testo in chiaro e il testo cifrato saranno vettori di h elementi:

$$k = (k_1, \ldots, k_h)\,,\ p = (p_1, \ldots, p_h)\,,\ c = (c_1, \ldots, c_h)$$

Le regole di cifratura e decifratura sono le stesse del cifrario di Cesare, applicate su ciascun elemento del vettore p o c:

$$\begin{aligned} c &= E_k(p) = (p_1 + k_1 \bmod n, \ldots, p_h + k_h \bmod n) \\ p &= D_k(c) = (c_1 - k_1 \bmod n, \ldots, c_h - k_h \bmod n) \end{aligned}$$

In questo cifrario gli scorrimenti per ciascun elemento del blocco dipendono dalla posizione al suo interno; la stessa lettera verrà cifrata in modo diverso a seconda della sua posizione nel blocco; l'analisi di frequenza si applica, suddividendo la frequenza di un simbolo su h possibili modi di cifrarlo (la difficoltà cresce all'aumentare di h).

2.3. Cifrario a permutazione

Si tratta di un cifrario a blocco (lavora su h simboli alla volta), che permuta le posizioni dei caratteri in ogni blocco. Al contrario del cifrario a sostituzione, la permutazione in ogni blocco è indipendente. La chiave adottata è una permutazione delle posizioni all'interno del blocco.

Esempio 2.1. *Dato il cifrario a permutazione con blocchi di dimensione $n = 6$ e permutazione $k = (3, 5, 1, 6, 4, 2)$, cifrare il messaggio $p =$ "she sells sea shells" (gli spazi tra le parole non sono caratteri dell'alfabeto).*

✓Identifichiamo i blocchi di $n = 6$ caratteri; il primo blocco da sinistra è $\mathbb{B}_1 =$ "shesel", che possiamo indicare come:

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \text{s}, & \text{h}, & \text{e}, & \text{s}, & \text{e}, & \text{l} \end{pmatrix}$$

A questo punto applichiamo la permutazione k, che sposta i simboli del blocco nella seguente configurazione:

$$c = \begin{pmatrix} 3 & 5 & 1 & 6 & 4 & 2 \\ \text{e}, & \text{e}, & \text{s}, & \text{l}, & \text{s}, & \text{h} \end{pmatrix}$$

□

Questo cifrario risulta poli-alfabetico, e ogni blocco di lunghezza n può essere cifrato in $n!$ modi differenti.

2.4. Proprietà fondamentali dei cifrari a blocchi

In un cifrario a blocco di lunghezza n, si prende un messaggio in chiaro e a partire da esso si ottiene un messaggio cifrato, entrambi della stessa lunghezza ($|p| = |c| = n$). Nell'articolo[1] sulla crittografia pubblicato da Shannon, egli espone due proprietà fondamentali che ogni sistema crittografico dovrebbe avere:

Diffusione: Si ha diffusione perfetta se, cambiando 1 bit del blocco in chiaro, tutti i bit del blocco cifrato corrispondente cambiano in maniera apparentemente casuale, con probabilità 1/2. La perfetta diffusione vale anche nel senso opposto: cambiando 1 bit del blocco cifrato, tutti i bit del blocco in chiaro corrispondente cambiano con probabilità 1/2.

Confusione: La chiave deve avere effetto su tutti i bit del blocco cifrato: a parità di blocco in chiaro, se cambia 1 bit della chiave, tutti i bit del blocco cifrato cambieranno, con probabilità 1/2.

Queste due proprietà si riassumono nella seguente affermazione:

Proposizione 2.1. *Non solo ogni bit della chiave ha effetto su tutti i bit del blocco cifrato, ma anche tutti i bit della chiave hanno effetto su ogni bit del blocco cifrato.*

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

CAPITOLO 3

Cifrario DES

3.1. Introduzione

3.1.1. Nascita del DES come standard. L'algoritmo Data Encryption Standard (DES) nasce negli anni '70 per soddisfare la necessità degli Stati Uniti di usare un cifrario per le comunicazioni segrete; allora l'NBS (attuale NIST) pubblica un bando, a cui risponde IBM con l'algoritmo LUCIFER.

L'algoritmo di IBM conteneva delle tabelle di sostituzione, che furono modificate dall'NBS: questo portò a pensare, per molti anni, che il governo introdotto una trapdoor nelle tabelle di sostituzione, tuttavia non fu mai trovata.

Infine il DES fu pubblicato come standard nel '77; col progredire della potenza di calcolo nei calcolatori, questo algoritmo è divenuto suscettibile a diversi attacchi, così sono state introdotte delle sue varianti (compatibilmente con l'hardware legacy), come il triplo DES (3DES).

Di per sé l'algoritmo DES è robusto, tuttavia la sua chiave di 56bit è relativamente corta ($|\mathscr{K}| = 2^{56} \simeq 2^6 \cdot 10^{15}$), e questo permette di effettuare in breve tempo un'analisi esaustiva dello spazio delle chiavi. Oltre a questo, DES sarebbe stato vulnerabile ad attacchi di crittoanalisi differenziale (si cerca un errore nella realizzazione della proprietà di perfetta diffusione, prendendo un blocco di testo in chiaro e cifrandolo, cambiando un bit alla volta e cercando evidenza statistica all'interno del blocco cifrato) se il suo algoritmo avesse avuto 14 *round*; tuttavia la specifica di IBM prevede 16 *round* (si veda 2.1).

DES è un cifrario a blocchi di 64bit, che nel modo più semplice lavora prendendo un blocco, cifrandolo e poi ripetendo il processo in modo indipendente per il blocco successivo (modo *ECB*).

3.1.2. Schema Feistel. Descriviamo l'idea dietro a questo schema con la procedura da implementare per realizzarlo:

- prendiamo un blocco di bit B_1, e dividiamolo in due metà, in modo da avere la sinistra L_1 e la destra R_1;
- usiamo la chiave K per generare una sotto-chiave di round K_i;
- permutiamo le due metà al round i-esimo, facendo passare la metà di sinistra attraverso una funzione fortemente non lineare che utilizzi la chiave: $B_i = R_{i-1}|L_{i-1} \oplus f(R_{i-1}, K_i)$;
- se vi sono n round in totale, non si effettua la permutazione di L_i ed R_i al round n-esimo, e si ottiene $B_n = R_n|L_n$

La notazione $A|B$ indica la concatenazione di B dopo di A

Possiamo presumere che, continuando a permutare le due metà del blocco, grazie alla funzione $f()$, vi sarà un'ottima diffusione dei bit in ingresso; inoltre la confusione sarà garantita dalla presenza delle chiavi di round K_i, generate a partire dalla chiave iniziale.

Lo schema Feistel permette di decifrare con la stessa procedura della cifratura, utilizzando B_n e K_n come ingresso, e ottenendo B_1 alla fine degli n round.

3.2. Algoritmo DES

Il DES applica 16 volte (round) il blocco di Feistel, utilizzando la seguente procedura:

(1) si applica una permutazione iniziale (IP) fissata al blocco di 64bit in chiaro;
(2) si prende la chiave DES (da 64bit), si rimuovono gli 8bit di parità e si utilizzano i 56bit per produrre le successive chiavi di round K_i;
   (a) si permutano i bit della chiave da 56bit, poi essa viene divisa in due blocchi da 28bit, rispettivamente C_0 e D_0;
   (b) si produce la chiave di round secondo una tabella di shift fissa (funzione LS_i), che fa scorrere i bit delle due metà della chiave (C_i, D_i) in modo circolare di 1 o 2 posizioni, a seconda del numero i del round: $C_i = LS_i(C_{i-1})$, $D_i = LS_i(D_{i-1}) \implies K_i = C_i|D_i$;
   (c) tramite una tabella di riduzione, si estraggono dalla chiave di round 48bit, da usare nella funzione $f()$.
(3) si prende la chiave di round K_i e il blocco sinistro di round R_{i-1} e si genera il blocco destro L_i tramite la funzione $f()$
   (a) il blocco R_{i-1} in ingresso viene espanso a 48bit tramite una tabella di espansione;
   (b) si somma modulo due la chiave di round al blocco espanso: $E(R_{i-1}) \oplus K_i$;
   (c) si divide il risultato da 48bit in 8 gruppi da 6bit;
   (d) ciascun gruppo viene modificato tramite una S-Box (8 in totale), che prende un blocco da 6bit e lo riduce a 4bit;
   (e) gli 8 gruppi da 4bit sono uniti in un blocco da 32bit, che viene permutato;
   (f) il blocco ottenuto viene infine usato assieme alla chiave nella funzione $f()$.

Initial Permutation (IP). Essa lavora su un blocco di 64bit, e produce un'altro blocco altrettanto lungo. Osserviamo che la permutazione iniziale, nota e fissa, non migliora diffusione o confusione; alcune ipotesi sostengono che poteva essere un modo per inizializzare l'hardware dell'epoca ('70).
La seguente tabella di look-up realizza la permutazione iniziale del DES:

Funzione f. Essa lavora su blocchi di 32bit, e garantisce ottima diffusione grazie alle permutazioni effettuate sui blocchi di bit.

S-Box. Tabella di sostituzione che prende un blocco da 6bit, e usando il primo e l'ultimo per indicare la riga e i 4 centrali per indicare la colonna, si ottiene un blocco da 4bit. Infatti, la tabella dell'S-Box contiene valori di massimo 4bit, che vengono indirizzati nel modo seguente: se $B_h = \overset{R_0}{0}\underset{C_0}{0}\underset{C_1}{1}\underset{C_2}{0}\underset{C_3}{0}\overset{R_1}{1}$, allora il valore ottenuto sarà l'elemento alla riga $01_2 = 1_{10}$ e alla colonna $0100_2 = 4_{10}$ della S-Box h (righe e colonne sono numerate a partire da 0); queste tabelle sono state progettate per realizzare una trasformazione non lineare sui blocchi.

Chiave di round. Ogni bit della chiave sarà utilizzato mediamente in 14 su 16 round; questo permette di realizzare ottima confusione. I bit di parità della chiave, che portano la sua lunghezza da 56 a 64bit, sono un metodo per controllarne l'integrità (per esempio per trasmettere la chiave senza errori, come controllo aggiuntivo).
Notiamo che il DES rispetta il principio di Kerchoffs (pagina 40) infatti la chiave deve rimanere privata, ed è l'unico "ingrediente" che permette di decifrare un blocco

cifrato; l'algoritmo è invece pubblico e ben definito. L'algoritmo è infatti progettato per rendere l'analisi esaustiva dello spazio delle chiavi l'unico tipo di attacco possibile.

3.3. Modi operativi

I cifrari a blocchi, come DES, supportano i modi operativi descritti nelle seguenti sotto-sezioni.

Electronic Code Book (ECB). Il messaggio in chiaro è diviso in blocchi, i quali sono cifrati in modo indipendente, utilizzando la stessa chiave:

$$c_i = E_k(p_i), \quad p_i = E_k(c_i)$$

Risulta facile individuare i blocchi ripetuti, inoltre è possibile costruire una tabella con le corrispondenze tra blocchi i chiaro e blocchi cifrati.

Cipher Block Chaining (CBC). Si tratta di un modo operativo concatenato: in qualche modo, ciascun blocco è legato agli altri, e si ottiene l'effetto di non avere blocchi ripetuti (questo vale per ciascun modo concatenato). CBC usa un vettore di inizializzazione (IV), come se fosse una chiave addizionale, come primo blocco cifrato c_0; a partire da esso, le funzioni di cifratura e decifratura sono definite come:

$$c_i = E_k(p_i \oplus c_{i-1}), \quad p_i = c_{i-1} \oplus D_k(c_i)$$

Si noti che IV può essere inviato anche in chiaro.

Cipher Feedback (CFB). Si tratta di un modo concatenato. Consiste nel cifrare il vettore di inizializzazione e poi sommarlo al blocco in chiaro; il risultato è usato come IV per il round successivo. Questo algoritmo crea una sequenza di bit pseudo-casuali.

Per decifrare non è necessaria una funzione specifica, ma basta ripetere la somma in base 2 con i blocchi cifrati al posto dei blocchi in chiaro:

$$c_i = p_i \oplus E_k(c_{i-1}), \quad p_i = c_i \oplus E_k(c_{i-1})$$

Lo svantaggio di questo cifrario è l'amplificazione degli errori di trasmissione, da un blocco al suo successivo.

Output Feedback Mode (OFB). Si tratta di un modo concatenato. Consiste nel cifrare il vettore di inizializzazione, sommarlo al blocco in chiaro, e poi usare l'IV cifrato come ingresso per il blocco successivo. La catena di cifrature effettuate sull'IV costituisce una PRBS (pseudo-random binary sequence, una sequenza pseudo-casuale di bit). L'unico modo per generare la stessa sequenza è conoscere la chiave; inoltre un errore di trasmissione non si propaga tra i blocchi.

Le funzioni di cifratura e decifratura sono definite come:

$$c_i = p_i \oplus E_k^{(i)}(\mathrm{IV}), \quad p_i = c_i \oplus E_k^{(i)}(\mathrm{IV})$$

dove $E_k^{(i)}(\mathrm{IV})$ è la funzione E_k applicata in modo ricorsivo per i volte all'IV.

Counter Mode (CTR). Questo modo non è concatenato. Consiste nel cifrare un IV con la chiave, per poi sommarlo al blocco in chiaro; per ciascun blocco in chiaro i-esimo viene usato un $\mathrm{IV}^{(i)} := \mathrm{IV} + i$, e ogni blocco cifrato è prodotto in modo indipendente; la sicurezza di questo modo si basa unicamente su quella della funzione di cifratura (in caso di ottima diffusione e confusione, un attaccante non può sapere se a IV sta venendo sommato un intero e quale esso sia).

LE funzioni di cifratura e decifratura sono definite come:

$$c_i = p_i \oplus E_k\left(\mathrm{IV}^{(i)}\right), \quad p_i = c_i \oplus E_k\left(\mathrm{IV}^{(i)}\right)$$

3.4. Sicurezza del DES

3.4.1. Doppio DES. Per aumentare la sicurezza di DES potremmo provare ad utilizzare una chiave più lunga; in tal caso, ci chiediamo se esista una chiave $K' = K_1|K_2$ tale che $|K'| = 56 + 56 = 112\text{bit}$, e per essa si verifichi che $E_{K'}(p) \equiv E_{K_2}(E_{K_1}(c))$.

Se prendiamo la funzione di cifratura *affine* (si veda la sezione 2.2.2), dove abbiamo $K_1 = (a, b)$, allora possiamo cifrare un testo in chiaro p come $c_1 = E_{K_1}(p) = a \cdot p + b \bmod n$; prendiamo un'altra chiave $K_2 = (c, d)$ e con essa cifriamo di nuovo il messaggio p, ottenendo $c_2 = E_{K_2}(p) = c \cdot p + d \bmod n$.

Si nota subito che, per le caratteristiche della funzione di cifratura (combinazione lineare con la chiave), esiste $K' = (a \cdot c, b + d)$ per cui vale $c' = E_{K_2}(E_{K_1}(p)) \equiv E_{K'}(p) = ac \cdot p + b + d \bmod n$. Dato che il cifrario affine forma un gruppo algebrico (chiuso rispetto alla composizione), cifrando due volte con due chiavi K_1 e K_2 non viene aumentata la cardinalità dello spazio delle chiavi, dato che esiste in ogni caso una chiave equivalente K' di lunghezza identica alle singole chiavi.

Il DES non gode di questa proprietà: cifrando più volte il messaggio in chiaro con due chiavi differenti, aumenta di $2^{|K|}$ la cardinalità dello spazio delle chiavi. Nel caso del doppio DES (2DES), si ha una chiave $K = K_1|K_2$ lunga il doppio di una chiave singola, dunque $|\mathscr{K}| = 2^{112} \simeq 2^2 \cdot 10^{33}$.

Il doppio DES è suscettibile dell'attacco Meet-in-the-Middle (MiM): questa vulnerabilità permette di analizzare al più 2^{57} chiavi, invece delle 2^{112} totali; per ovviare al problema si è scelto di proseguire con la composizione, ottenendo il triplo DES.

3.4.2. Triplo DES. Questo cifrario è la composizione di tre chiavi DES, che portano la cardinalità dello spazio delle chiavi a 2^{168}; anche con un attacco MiM lo spazio delle chiavi si riduce al più a 2^{112}, e al giorno d'oggi è considerato sicuro almeno quanto l'AES, ma è meno veloce (bisogna cifrare 3 volte con DES).

Il metodo più intuitivo per implementare il 3DES è effettuare tre volte la cifratura con tre chiavi differenti: $c = E_{K_1}(E_{K_2}(E_{K_3}(p)))$; un approccio che utilizza solo due chiavi, senza ridurre lo spazio delle chiavi, è il seguente:

$$c = E_{K_1}(D_{K_2}(E_{K_1}(p))), \quad p = D_{K_1}(E_{K_2}(D_{K_1}(c)))$$

Questa implementazione del triplo DES è compatibile con l'hardware per il singolo DES.

Rivest ha proposto la seguente alternativa, chiamata DESX, che usa tre chiavi ma effettua una sola cifratura DES, ed è sicura quanto il 3DES standard:

$$c = K_3 \oplus E_{K_2}(K_1 \oplus p)$$

3.4.3. Attacco Meet-in-the-Middle. Abbiamo già osservato che DES non è un gruppo; ci chiediamo come possiamo affrontare questa caratteristica, per ridurre lo spazio delle chiavi nel caso di una composizione. Conduciamo un attacco di testo in chiaro noto, partendo da una coppia corrispondente di testo in chiaro e testo cifrato $p \leftrightarrows c$, senza conoscere la chiave.

Poniamoci nel caso di un testo cifrato ricavato tramite 2DES come $c = E_{K_2}(E_{K_1}(p))$, e vogliamo ottenere $K' = K_1|K_2$. Se quanto affermato sul testo cifrato è vero, allora applicando a c la decifratura con K_2 si ottiene

$$D_{K_2}(c) = E_{K_1}(p) \tag{3.4.1}$$

Effettuiamo una esplorazione esaustiva dello spazio delle chiavi K_1 memorizzando i risultati in una tabella, per cui sarà necessario memorizzare 2^{56} valori da 8Byte (64bit) ovvero circa 0.5EB; poi effettuiamo la stessa analisi per lo spazio delle chiavi K_2, questa volta evitando di memorizzare i risultati ma confrontando ciascuna chiave trovata con i valori di K_1 salvati.

Notiamo che saranno necessarie al più $2 \cdot 2^{56} = 2^{57}$ operazioni di cifratura e decifratura con DES, e al più 2^{112} confronti.

3.5. Sicurezza delle password

3.5.1. Funzioni di hash. Per ragioni di sicurezza, una password non viene mai memorizzata in chiaro; in pratica, si usa una funzione di *hash*. Si tratta di una funzione con le seguenti caratteristiche:

- accetta un ingresso di lunghezza arbitraria, produce una uscita di lunghezza fissa;
- non è invertibile (ci sono infiniti messaggi con lo stesso hash in uscita dalla funzione);
- sia unidirezionale (presa una uscita $y(x)$, sia impossibile trovare uno qualunque degli infiniti x che produca quell'uscita);
- goda di ottima diffusione.

Usando una simile funzione, possiamo memorizzare lo hash della password, e confrontarlo con l'hash calcolato sulla password quando fornita.

È possibile attaccare queste funzioni provando un gran numero di ingressi casuali differenti, rispetto al numero delle possibili uscite; la sicurezza viene determinata dalla lunghezza dell'uscita.

Si può mitigare questo attacco scegliendo una funzione di hash particolarmente lenta.

3.5.2. Attacco del vocabolario. Dato che quasi nessuno sceglie una stringa veramente casuale come password, è possibile attaccare l'hash corrispondente generando tutti gli hash di tutte le parole di una lingua, apportando eventuali variazioni sulle singole parole (aggiunta di un numero, iniziale maiuscola, eccetera...) e poi confrontando gli hash ottenuti con quelli nel file delle password.

All'interno di sistemi con un gran numero di utenti, la probabilità di trovare una password "banale" per almeno qualche utenza è molto elevata.

La contromisura migliore è applicare un *salt* ("sale") alla password; esso è una stringa di bit, memorizzata in chiaro nel file delle password. L'hash "salato" è calcolato sulla stringa ottenuta dalla concatenazione del sale con la password.

Il sale non aumenta la complessità dell'attacco sul singolo utente, bensì ha il vantaggio che, nel caso di un attacco del vocabolario su numerose utenze, essi abbiano tutti un hash diverso, anche a parità di password, per via del sale (che sarà diverso per

ogni utente). Un attaccante dovrà calcolare l'hash per ogni parola del vocabolario, per ogni sale possibile.

Nei sistemi UNIX, le password sono salate con un salt a 12bit, e la funzione di hash è basata su DES; tradizionalmente si troncava la password utente a 64bit (8 caratteri), da questi si ottenevano i primi 7 bit di ciascun Byte (per rappresentare solo i caratteri stampabili), e i 56bit così ottenuti erano usati come chiave DES. Questa chiave K_h è poi usata per cifrare un blocco di 0 da 64bit, con 25 round di DES; a ogni round, i 12bit del sale sono usati per perturbare la funzione di cifratura (modificando le S-Box, rende inutile l'implementazione hardware a priori). Infine sale e blocco cifrato sono codificati in base 64.

La funzione così realizzata è reputata unidirezionale: infatti ottenere la password senza conoscere la chiave e con un DES da più di 16 round, per giunta perturbato dal sale, è un problema difficile.

Esercizi

Equazioni congruenziali

Esercizio 3.1. Risolvere la seguente equazione congruenziale:

$$28x \equiv 16 \pmod{412}$$

▶ *Esercizio: equazione congruenziale, modulo composto*

Soluzione 3.1. Procediamo calcolando $\text{MCD}(28, 412) = 4$; allora possiamo dividere per 4 tutti i termini dell'equazione, ottenendo

$$7x \equiv 4 \pmod{103}$$

L'equazione originale avrà 4 soluzioni, e la prima si ricava dall'equazione ridotta appena trovata:

$$x_0 \equiv 4 \cdot 7^{-1} \pmod{103}$$

Per calcolare l'inverso di 7 modulo 103 possiamo usare il Teorema 1.3 oppure l'algoritmo (1.3.4).

Teorema piccolo di Fermat
Il teorema afferma che, in questo caso, vale

$$7^{-1} \equiv 7^{101} \pmod{103}$$

dato che 103 è primo; adesso effettuiamo l'operazione di $7^{101} \bmod 103 = 59$ (possiamo usare l'algoritmo descritto ne la sezione §1.6), da cui segue che

$$7^{-1} \equiv 59 \pmod{103}$$

Algoritmo di Euclide Esteso
Usiamo l'algoritmo per ottenere che $7 \perp 103$ e in tal caso anche 7^{-1}; cominciamo a costruire la sequenza (1.3.2):

$$\begin{array}{ll} 103 = 14 \cdot 7 + 5 & q_1 = 14 \\ 7 = 1 \cdot 5 + 2 & q_2 = 1 \\ 5 = 2 \cdot 2 + \boxed{1} & q_3 = 2 \\ 2 = 2 \cdot 1 + 0 & q_4 = 1 \end{array}$$

Otteniamo che $\text{MCD}(7, 103) = 1$, e per quanto riguarda l'inverso costruiamo la sequenza (1.3.4):

$$\begin{array}{c} x_0 = 0 \\ x_1 = 1 \\ x_2 = -q_1 x_1 + x_0 = -14 \\ x_3 = -q_2 x_2 + x_1 = 15 \\ x_4 = -q_3 x_3 + x_2 = \boxed{-44} \end{array}$$

Segue che $-44 \bmod 103 = 59 = 7^{-1}$.

Ora che abbiamo l'inverso di 7 in modulo 103, siamo in grado di scrivere la soluzione dell'equazione ridotta:

$$x_0 \equiv 4 \cdot 59 \,(\mathrm{mod}\, 103)$$
$$x_0 \equiv 30 \,(\mathrm{mod}\, 103)$$

Le soluzioni successive saranno altre tre, a 103 di distanza da x_0, esse inoltre saranno in modulo 412:

$$X = \{30,\, 133,\, 236,\, 339\} \bmod 412$$

□

▶ *Esercizio: sistema di 2 congruenze*

ESERCIZIO 3.2. Trovare la congruenza equivalente al seguente sistema di congruenze:

$$\begin{cases} x \equiv 5 \,(\mathrm{mod}\, 11) \\ x \equiv 2 \,(\mathrm{mod}\, 20) \end{cases}$$

SOLUZIONE 3.2. Esisterà una sola congruenza equivalente, per il Teorema 1.2, infatti vale $11 \perp 20$; per riassumere le due congruenze in una sola, cerchiamo un valore x della successione $5 + k \cdot 11$ che sia anche congruente a 2 in modulo 20:

$$x = 5 + k \cdot 11 \equiv 2 \,(\mathrm{mod}\, 20)$$

Questo sarà vero per un valore di k pari a

$$k \cdot 11 \equiv 2 - 5 \,(\mathrm{mod}\, 20)$$
$$k \equiv -3 \cdot 11^{-1} \,(\mathrm{mod}\, 20)$$

Troviamo l'inverso di 11 in modulo 20 usando il Teorema 1.4:

$$11^{-1} \equiv 11^{\varphi(20)-1} \,(\mathrm{mod}\, 20)$$

sapendo che la funzione toziente di 20 vale $\varphi\,(20) = (2^2 - 2^1)\,(5^1 - 5^0) = 2 \cdot 4 = 8$, e che $11^{8-1} \bmod 20 = 11$, abbiamo la seguente congruenza per k:

$$k \equiv -33 \,(\mathrm{mod}\, 20)$$
$$k \equiv 7 \,(\mathrm{mod}\, 20)$$

Ritornando alla congruenza per x, si ottiene dalla precedente (con $k = 7$):

$$x = 5 + 77 \equiv 2 \,(\mathrm{mod}\, 20 \cdot 11)$$
$$x \equiv 82 \,(\mathrm{mod}\, 220)$$

dove abbiamo usato il Teorema 1.2 per comporre la congruenza in modulo 220. □

▶ *Esercizio: sistema di 3 congruenze*

ESERCIZIO 3.3. Trovare il valore della variabile x, che compare nel sistema di congruenze:

$$\begin{cases} x \equiv 1 \,(\mathrm{mod}\, 10) \\ x \equiv 2 \,(\mathrm{mod}\, 11) \\ x \equiv 0 \,(\mathrm{mod}\, 3) \end{cases}$$

SOLUZIONE 3.3. Mettiamo assieme la prima congruenza con la seconda, ottenendo:

$$x = 1 + k \cdot 10 \equiv 2 \,(\mathrm{mod}\, 11)$$

Risolviamo la precedente rispetto a k:

$$k \cdot 10 \equiv 1 \,(\mathrm{mod}\, 11)$$

Notando che $100 = 9 \cdot 11 + 1 \implies 100 \bmod 11 = 1$ possiamo affermare subito che $k = 10$; per il Teorema 1.2 scriviamo:

$$x \equiv 101 \,(\mathrm{mod}\, 110)$$

Unendo quanto ottenuto con la terza congruenza del sistema, si ha:

$$x = 101 + k \cdot 110 \equiv 0 \,(\mathrm{mod}\, 3)$$

Effettuando l'operazione di modulo 3 sui termini dell'equazione, avremo:

$$x = 2 + h \cdot 2 \equiv 0 \,(\mathrm{mod}\, 3)$$

Risolviamo la precedente rispetto ad h:

$$2 \cdot h \equiv -2 \,(\mathrm{mod}\, 3)$$
$$h \equiv 1 \cdot 2^{-1} \,(\mathrm{mod}\, 3)$$

Osserviamo che l'inverso di 2 in $\mathbb{Z}_3$ è proprio 2, infatti $2 \cdot 2 = 4 \,\mathrm{mod}\, 3 = 1$; allora h è congruente a 2 in modulo 3, e sostituendo nella congruenza di x otteniamo:

$$x \equiv 101 + 220 \,(\mathrm{mod}\, 110 \cdot 3)$$
$$x \equiv 321 \,(\mathrm{mod}\, 330)$$

Abbiamo trovato che x deve valere $321 + k \cdot 330$ per generare le tre congruenze esaminate. □

Elementi primitivi

ESERCIZIO 3.4. Quanti elementi primitivi ha l'insieme $\mathbb{Z}_{31}^*$, e quali sono?

► *Esercizio: elementi primitivi di un insieme dei residui*

SOLUZIONE 3.4. Gli elementi primitivi sono in numero $\varphi(31-1) = \varphi(30) = (2^1 - 2^0)(3^1 - 3^0)(5^1 - 5^0) = 1 \cdot 2 \cdot 4 = 8$ (applichiamo una delle formule a pagina 15 per calcolare il toziente $\varphi(\circ)$).

Per capire quali siano gli elementi primitivi, usiamo il test (1.9.4) con i quozienti $q_i = \{2, 3, 5\}$, per ciascun elemento di $\mathbb{Z}_{31}^*$ (da 1 a 30); facendo le prove col test rispetto ai tre quozienti, si ottengono i seguenti elementi primitivi:

$$\alpha = \{3, 11, 12, 13, \ldots\}$$

□

ESERCIZIO 3.5. Trovare gli elementi primitivi dell'insieme $\mathbb{Z}_{13}^*$

► *Esercizio: elementi primitivi di un insieme dei residui*

SOLUZIONE 3.5. Usiamo un secondo metodo, a partire da un elemento sicuramente primitivo $\alpha \in \mathbb{Z}_{13}^*$; proviamo con 2, sottoponendolo al test (1.9.4):

$$2^6 \equiv -1 \,(\mathrm{mod}\, 13)\,, \quad 2^4 \equiv 3 \,(\mathrm{mod}\, 13)$$

Allora $\alpha = 2$ è un elemento primitivo rispetto a $p = 13$; possiamo elevare α a tutte le potenze, per ottenere tutti gli elementi dell'insieme. Tuttavia solo le potenze positive pari restituiranno un altro elemento primitivo rispetto a p che è un quadrato:

$$\begin{array}{lll}
\alpha^1 \equiv 2 \,(\mathrm{mod}\, 13) & \alpha^5 \equiv 6 \,(\mathrm{mod}\, 13) & \alpha^9 \equiv 5 \,(\mathrm{mod}\, 13) \\
\alpha^2 \equiv \boxed{4} \,(\mathrm{mod}\, 13) & \alpha^6 \equiv \boxed{12} \,(\mathrm{mod}\, 13) & \alpha^{10} \equiv \boxed{10} \,(\mathrm{mod}\, 13) \\
\alpha^3 \equiv 8 \,(\mathrm{mod}\, 13) & \alpha^7 \equiv 11 \,(\mathrm{mod}\, 13) & \alpha^{11} \equiv 7 \,(\mathrm{mod}\, 13) \\
\alpha^4 \equiv \boxed{3} \,(\mathrm{mod}\, 13) & \alpha^8 \equiv \boxed{9} \,(\mathrm{mod}\, 13) & \alpha^{12} \equiv \boxed{1} \,(\mathrm{mod}\, 13)
\end{array}$$

Le radici primitive in $\mathbb{Z}_{13}^*$ sono:

$$a_q = \{1, 3, 4, 9, 10, 12\}$$

Si noti che ci troviamo nel caso $p \equiv 1 \,(\mathrm{mod}\, 4)$, per cui se non esiste $\sqrt{a}$ nemmeno $\sqrt{-a}$ è definita rispetto a p. □

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).

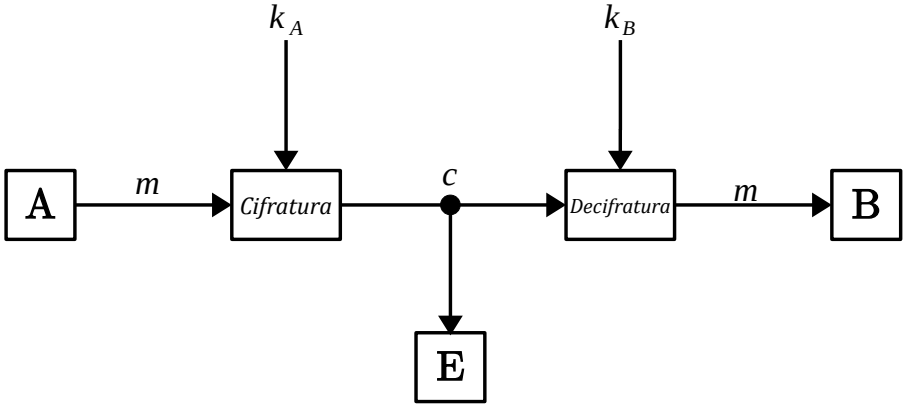

Figura A.0.1. Scenario fondamentale di comunicazione

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