Esercitazioni Campi Finiti

Corso di Algebra e Logica 2

Gerardo Pelosi

Dipartimento di Elettronica e Informazione
Politecnico di Milano
email: pelosi@elet.polimi.it

1 Algoritmo di Euclide

Sia D un dominio agli ideali principali, cioè un anello dotato di unità in cui vale la legge di annullamento del prodotto e in cui ogni ideale è principale. Esempio: $D =< \mathbb{Z}, +, \cdot >$, $D =< \mathbb{F}[X], +, \cdot >$

Definizione 1.1. Dati $a, b \in D$, si definisce Massimo Comun Divisore $d = MCD(a, b)$, l'elemento $d \in D$ tale che $d|a$, $d|b$ e $\forall y \in D : y|a \wedge y|b \Rightarrow y|D$.

Senza pretesa di rigore matematico, ci limitiamo a ricordare come, dati $a, b \in D$ se $d \in D$ è il loro massimo comun divisore allora d divide anche una loro qualunque combinazione lineare: $d|x \cdot a + y \cdot b$ con $x, y \in D$. Si può dimostrare, assieme all'esistenza di d anche l'esistenza di almeno una coppia di elementi $x_a, y_b \in D$ tali che

$$d = x_a a + y_b b$$

Il precedente risultato permetterà di calcolare l'inverso moltiplicativo in un campo finito.

Lemma 1.1. *$a, b \in D$; $a > b$; se D è un dominio euclideo(es.: $\mathbb{Z}, \mathbb{F}[X]$) si può definire una nozione di quoziente e si ha $q = \lfloor a/b \rfloor$; $r = a \bmod b = a - qb \in \{0, 1, 2 \ldots, b-1\}$ e vale il seguente:*

$$MCD(a, b) = MCD(b, a \bmod b)$$

Usando il lemma precedente si possono scrivere le seguenti uguaglianze:

$a > b > 0$	$d = MCD(a, b)$
$r_0 = a$	
$r_1 = b$	$d = MCD(r_0, r_1)$
$r_2 = r_0 \bmod r_1 = r_0 - \lfloor r_0/r_1 \rfloor r_1;\ \ 0 \leq r_2 < r_1$	$d = MCD(r_1, r_2)$
$r_3 = r_1 \bmod r_2 = r_1 - \lfloor r_1/r_2 \rfloor r_2;\ \ 0 \leq r_3 < r_2$	$d = MCD(r_2, r_3)$
$r_4 = r_2 \bmod r_3 = r_2 - \lfloor r_2/r_3 \rfloor r_3;\ \ 0 \leq r_4 < r_3$	$d = MCD(r_3, r_4)$
$\ldots$	$\ldots$
$r_n = r_{n-2} \bmod r_{n-1} = r_{n-2} - \lfloor r_{n-1}/r_{n-2} \rfloor r_{n-2};\ \ r_n = 0$	$d = MCD(r_{n-1}, 0)$

per un certo intero n si ha:

$$d = MCD(a, b) = r_{n-1}$$

Pensiamo ora di riscrivere tutti i passaggi precedenti come combinazioni lineari di a, b:

$$
\begin{aligned}
&a > b > 0\\
&r_0 = 1 \cdot a + 0 \cdot b\\
&r_1 = 0 \cdot a + 1 \cdot b\\
&r_2 = r_0 \, mod \, r_1 = (1a + 0b) - \lfloor \tfrac{r_0}{r_1} \rfloor (0a + 1b) = (\xi_2 a + \eta_2 b);\ 0 \leq r_2 < r_1\\
&r_3 = r_1 \, mod \, r_2 = (0a + 1b) - \lfloor \tfrac{r_1}{r_2} \rfloor (\xi_2 a + \eta_2 b) = (\xi_3 a + \eta_3 b);\ 0 \leq r_3 < r_2\\
&r_4 = r_2 \, mod \, r_3 = (\xi_2 a + \eta_2 b) - \lfloor \tfrac{r_2}{r_3} \rfloor (\xi_2 a + \eta_2 b) = (\xi_4 a + \eta_4 b);\ 0 \leq r_4 < r_3\\
&\dots\\
&r_{n-1} = r_{n-3} \, mod \, r_{n-2} = (\xi_{n-3} a + \eta_{n-3} b) - \lfloor \tfrac{r_{n-3}}{r_{n-2}} \rfloor (\xi_{n-2} a + \eta_{n-2} b) =\\
&\qquad = (\xi_{n-1} a + \eta_{n-1} b);\ 0 \leq r_{n-1} < r_{r-2}\\
&r_n = r_{n-2} \, mod \, r_{n-1} = (\xi_{n-2} a + \eta_{n-2} b) - \lfloor \tfrac{r_{n-3}}{r_{n-2}} \rfloor (\xi_{n-1} a + \eta_{n-1} b) =\\
&\quad = (\xi_n a + \eta_n b);\ r_n = 0
\end{aligned}
$$

per cui:

$$d = r_{n-1} = \xi_{n-1} a + \eta_{n-1} b;\ \ \xi, \eta \in D$$

Formalizzando la precedente costruzione si ottiene l'algoritmo di Euclide esteso per il calcolo del massimo comun divisore.

Algorithm 1.1: Algoritmo esteso di Euclide

Input: $a, b \in D$
Output: $d = \xi_a \cdot a + \eta_b \cdot b,\ \xi_a, \eta_b$

```
1  begin
2      u ← (a, 1, 0)
3      v ← (b, 0, 1)
4      repeat
5          w ← u − ⌊u[0]/v[0]⌋ · v
6          u ← v
7          v ← w
8      until (w[0] = 0)
9      d ← u[0], ξ_a ← u[1], η_b ← u[2]
10     return (d, ξ_a, η_b)
11 end
```

Esempio 1.1. *Sia* $D =< Z, +, \cdot >$

$d = MCD(11, 5) = 11\xi + 5\eta$*;*

$\underline{u} \leftarrow (11, 1, 0)$*;*
$\underline{v} \leftarrow (5, 0, 1)$*;*

$q = \lfloor \frac{11}{5} \rfloor = 2$*,* $\underline{w} \leftarrow (11 - 2 \cdot 5, 1 - 0 \cdot 2, 0 - 1 \cdot 2) = (1, 1, -2)$*;*
$\underline{u} \leftarrow (5, 0, 1)$*;*
$\underline{v} \leftarrow (1, 1, -2)$*;*

$q = \lfloor \frac{5}{1} \rfloor = 5$*,* $\underline{w} \leftarrow (5 - 1 \cdot 5, 0 - 1 \cdot 5, 1 - (-2) \cdot 5) = (0, -5, 11)$*;*
$\underline{u} \leftarrow (1, 1, -2)$*;*
$\underline{v} \leftarrow (0, -5, 11)$*;*

$d = 1$*;* $\xi = 1$*;* $\eta = -2$*.*

infatti: $1 = 1 \cdot 11 + (-2) \cdot 5$*.*

2 Esercizi sui campi Z_p

2.1 Esercizio 1

Si consideri il seguente campo finito: $\mathbb{F}_7 \cong \mathbb{Z}/7\mathbb{Z}$*.*

(1) Determinare le tavole di addizione e di moltiplicazione.

$+$	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

$\cdot$	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

(2) Determinare il numero degli elementi generatori del campo.

Se $n = |\mathbb{F}_7^*|$ è la cardinalità del gruppo, allora il numero di elementi generatori è: $\varphi(n) = \varphi(6) = 2$.

N.B.:

Proposizione 2.1. *In un gruppo ciclico* $G =< g >$ *di ordine* $n = |g|$*, dato* $h \in \mathbb{N}$ *si ha:*

$$|g^h| = \frac{n}{MCD(n,h)}$$

Dim. Posto $r = |g^h|$, si ha che $g^{hr} = (g^h)^r = e \Rightarrow n|hr$; questo vuol dire che esiste un intero m tale che $rh = mn$; dividendo ambo i membri per $MCD(n,h)$ si ottiene:

$$r\frac{h}{MCD(n,h)} = m\frac{n}{MCD(n,h)}$$

$\frac{h}{MCD(n,h)}$ e $\frac{n}{MCD(n,h)}$ sono co-primi per costruzione, quindi si può concludere:

$$\frac{n}{MCD(n,h)}|r$$

Viceversa si può osservare che

$$(g^h)^{\frac{n}{MCD(n,h)}} = (g^n)^{\frac{h}{MCD(n,h)}} = e \Rightarrow r|\frac{n}{MCD(n,h)}$$

da cui la tesi. □

Teorema 2.1. *Dato* $n \in \mathbb{N}^+$*, la funzione* $\tau(n)$*, definita come il numero dei divisori positivi di* n*:*

$$\tau(n) = |\{d \in \mathbb{N}^+ : d|n\}| = \sum_{d|n} 1$$

è una funzione moltiplicativa. Dati $n, m \in \mathbb{N}^+$ *tali che* $MCD(n,m) = 1$ *allora*

$$\tau(n \cdot m) = \tau(n) \cdot \tau(m)$$

Il gruppo $\mathbb{F}_7^*$ è ciclico, quindi ammete almeno un generatore $g_0 = g \in \mathbb{F}_7^*$, cioè: $< g_0 >= \mathbb{F}_7^*$.
Gli altri generatori saranno gli elementi $g_i = g_0^i$ tali che $MCD(n,i) = 1$ con $i \in \{2, 3, ..., n-1\}$, il calcolo del numero di generatori coincide quindi con il valore della funzione di Eulero, valutata in n:

$$\varphi(n) = |\{i \in \mathbb{Z},\ 0 < i < n\ :\ MCD(n,i) = 1\}|$$

Valgono i seguenti:

Teorema 2.2. *Siano* $n, m \in \mathbb{N}^+$ *con* $n > m$*, se* $MCD(n,m) = 1$ *allora* $\varphi(n \cdot m) = \varphi(n) \cdot \varphi(m)$

Teorema 2.3. *Dato $n \in \mathbb{N}^+$, la funzione $\tau(n)$, definita come il numero dei divisori positivi di n:*

$$\tau(n) = |\{d \in \mathbb{N}^+ : d|n\}| = \sum_{d|n} 1$$

è una funzione moltiplicativa. Dati $n, m \in \mathbb{N}^+$ tali che $MCD(n, m) = 1$ allora

$$\tau(n \cdot m) = \tau(n) \cdot \tau(m)$$

Lemma 2.1. *Siano p un numero primo e $k \in \mathbb{N}^+$, allora*

$$\varphi(p^k) = p^k - p^{k-1}$$
$$\tau(p^k) = k + 1$$

Dim. Cerchiamo tutti i numeri interi minori di p^k che abbiano unn fattore in comune con p^k. Osserviamo che tali interi saranno tutti multipli di p, pertanto della forma $h \cdot p$ con $h \in \{1, \ldots, p^{k-1}\}$. Il numero di interi minori di p^k che hanno un fattore in comune con p^k è: p^{k-1}. Il numero di interi coprimi con p^k è dunque immediato: $\varphi(p^k) = p^k - p^{k-1}$.
Per dimostrare che $\tau(p^k) = k + 1$ basta osservare che i divisori di p^k sono: 1, p, $p^2, \ldots,$ p^k. □

Lemma 2.2. *Dato $n \in \mathbb{N}^+$ e la sua scomposizione in fattori primi $n = \prod_{j=0}^{s} p_j^{e_j}$ con $s, e_j \in \mathbb{N}$ allora*

$$\varphi(n) = \prod_{j=0}^{s} p_j^{e_j} - p_j^{e_j - 1} = n \prod_{j=0}^{s} (1 - \frac{1}{p_j}).$$

$$\tau(n) = \prod_{j=0}^{s} (1 + e_j).$$

Dim. È un'applicazione immediata delle proprietà enunciate precedentemente. □

(3) Determinare gli elementi generatori del campo.

Procedendo per tentativi, iniziamo a verificare se $g = 2$ è un generatore: $2^1 = 2$, $2^2 = 4$, $2^3 = 8 = 1$, quindi l'elemnto $g = 2$ ha ordine 3 e non è un generatore. Proviamo quindi con $g = 3$: $3^1 = 3$, $3^2 = 2 mod 7$, $3^3 = 6 mod 7 \neq 1$, osservando la fattorizzazione dell'ordine del gruppo $n = |\mathbb{F}_7^*| = 6 = 2 \cdot 3$ si deduce l'esistenza di soli elementi di ordine 1, 2, 3 o 6 e quindi dai calcoli fatti possiamo concludere, senza ulteriori conti, che $g_0 = 3$ è un elemento generatore del campo $\mathbb{F}_7$.
L'altro generatore sarà: $g_1 = g_0^5 = 3^5 = 5 \, mod \, 7$.

(4) Determinare l'inverso moltiplicativo dei seguenti elementi:

$$x = 4, \qquad y = 3$$

Abbiamo due modi per calcolare l'inverso di un elemento e precisamente:

- il Piccolo Teorema di Fermat: p primo, $a \in F_p^*$, $a^{-1} = a^{p-2} \, mod \, p$.
- l'Algoritmo Esteso di Euclide per il calcolo del massimo comun divisore: p primo, $a \in F_p^*$, essendo $MCD(p, a) = 1 = \xi \cdot p + \eta \cdot a$ per opportuni interi ξ, $\eta \in \mathbb{Z}$. Prendendo il modulo p di ambo i membri della precedente uguaglianza si trova come: $\eta \cdot a = 1 mod p \Rightarrow \eta \equiv a^{-1} mod p$.

Utilizzando il primo metodo si ha:

$$p = 7, \; x = 4, x^{-1} \, mod \, 7 \equiv_7 4^5 \equiv_7 (4^2 \cdot 4^2 \cdot 4) \equiv_7 2^2 \cdot 4 \equiv_7 \mathbf{2}$$
$$p = 7, \; y = 3, y^{-1} \, mod \, 7 \equiv_7 3^5 \equiv_7 (3^2 \cdot 3^2 \cdot 3) \equiv_7 2^2 \cdot 3 \equiv_7 \mathbf{5}$$

Utilizzando invece l'algoritmo di Euclide:

$$p = 7, \; x = 4 \Rightarrow \; 1 = MCD(7, 4) = 7\xi + 4\eta$$

$$\begin{cases} \underline{u} \leftarrow (7, 1, 0); \\ \underline{v} \leftarrow (4, 0, 1); \end{cases}$$
$$\begin{cases} q = \lfloor \frac{7}{4} \rfloor = 1, \; \underline{w} \leftarrow (7 - 4, 1 - 0, 0 - 1) = (3, 1, -1); \\ \underline{u} \leftarrow (4, 0, 1); \\ \underline{v} \leftarrow (3, 1, -1); \end{cases}$$
$$\begin{cases} q = \lfloor \frac{4}{3} \rfloor = 1, \; \underline{w} \leftarrow (4 - 3 \cdot 1, 0 - 1 \cdot 1, 1 - (-1) \cdot 1) = (1, -1, 2); \\ \underline{u} \leftarrow (3, 1, -1); \\ \underline{v} \leftarrow (1, -1, 2); \end{cases}$$
$$\begin{cases} q = \lfloor \frac{3}{1} \rfloor = 3, \; \underline{w} \leftarrow (3 - 1 \cdot 3, 1 - (-1) \cdot 3, -1 - 2 \cdot 3) = (0, 4, 7); \\ \underline{u} \leftarrow (1, -1, 2); \\ \underline{v} \leftarrow (0, 4, 7); \end{cases}$$

$$d = 1; \; \xi = -1; \; \eta = 2 \Rightarrow x^{-1} = 4^{-1} \, mod \, 7 = \mathbf{2}$$

$$p = 7,\ y = 3 \Rightarrow\ 1 = MCD(7,3) = 7\xi + 3\eta$$

$$\begin{cases} \underline{u} \leftarrow (7,1,0); \\ \underline{v} \leftarrow (3,0,1); \end{cases}$$

$$\begin{cases} q = \lfloor \frac{7}{3} \rfloor = 2,\ \underline{w} \leftarrow (7-6, 1-0, 0-2) = (1,1,-2); \\ \underline{u} \leftarrow (3,0,1); \\ \underline{v} \leftarrow (1,1,-2); \end{cases}$$

$$\begin{cases} q = \lfloor \frac{3}{1} \rfloor = 3,\ \underline{w} \leftarrow (3-1\cdot 3, 0-1\cdot 3, 1-(-2)\cdot 3) = (0,-3,7); \\ \underline{u} \leftarrow (1,1,-2); \\ \underline{v} \leftarrow (0,-3,7); \end{cases}$$

$$d = 1;\ \xi = -2;\ \eta = -2 \Rightarrow y^{-1} = 3^{-1}\ mod\ 7 \equiv_7 \mathbf{-2} \equiv_7 \mathbf{5}$$

2.2 Algoritmi di Esponenziazione

In questa sezione descriveremo due varianti dell'algoritmo più conosciuto per effettuare l'operazione di esponenziazione in maniera efficiente. Il problema è formulato nel modo seguente: dati $a, n \in \mathbb{N}$, si vuole calcolare l'intero $c = a^n$ effettuando un numero di moltiplicazioni minore di n.
Indicando con t il numero di cifre binarie necessarie per rappresentare l'intero n, cioè:

$$t = \lceil \lg_2 n \rceil,\ n = (n_{t-1}, \ldots, n_1, n_0) \text{ con } n_i \in \{0,1\},\ i \in \{0,1,\ldots,t-1\}$$

possiamo scrivere:

$$c = a^n = a^{\sum_{j=0}^{t-1} n_j 2^j} = a^{n_{t-1}2^{t-1} + n_{t-2}2^{t-2} + \ldots + n_1 2^1 + n_0} \tag{1}$$

A seconda del modo in cui è possibile leggere l'ultimo membro della catena di uguaglianze appena scritte si esibiscono diverse formalizzazioni dell'algoritmo noto come Square and Multiply(S&M).

2.2.1 Square and Multiply - Left to Right

Pensando di scandire l'esponente dell'eq. (1) da sinistra verso destra, vale la seguente uguaglianza:

$$c = a^n = ((\cdots((a^{n_{t-1}})^2 \cdot a^{n_{t-2}})^2 \cdots)^2 \cdot a^{n_1})^2 \cdot a^{n_0}$$

Esempio 2.1. *Si supponga di voler applicare il metodo precedente al calcolo di $c = 5^6$; allora $a = 5$, $t = 3$, $n = 6 = 110_2 = 1\cdot 2^2 + 1\cdot 2^1 + 0\cdot 2^0$, si può scrivere:*

$$c = 5^{110_2} = ((5^1)^2 \cdot 5^1)^2 \cdot 5^0 = (5^2 \cdot 5)^2 = 15625.$$

Il costo computazionale del metodo appena illustrato, espresso in numero di moltiplicazioni e quadrati necessari per portare a termine la computazione è uguale a: $\lceil \lg_2 n \rceil$ quadrati + $\frac{1}{2}\lceil \lg_2 n \rceil$ moltiplicazioni (caso medio).

Algorithm 2.1: S&M Left to Right

Input: a, n, $t = \lceil \lg_2 n \rceil$, $n = (n_{t-1}, \ldots, n_1, n_0), n \geq 0$
Output: $c = a^n$

```
1  begin
2      if n = 0 then
3          return 1
4      c ← a
5      for i ← t − 2 down-to 0 do
6          c ← c^2
7          if n_i = 1 then
8              c ← c · a
9      return c
10 end
```

2.2.2 Square and Multiply - Right to Left

Pensando di scandire l'esponente dell'eq. (1) da destra verso sinistra, vale la seguente uguaglianza:

$$c = a^n = (a^{2^0})^{n_0} \cdot (a^{2^1})^{n_1} \cdot (a^{2^2})^{n_2} \cdots (a^{2^{t-1}})^{n_{t-1}}$$

Esempio 2.2. *Si supponga di voler applicare il metodo precedente al calcolo di* $c = 5^6$*; allora* $a = 5$, $t = 3$, $n = 6 = 110_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$*, si può scrivere:*

$$c = 5^{110_2} = (5^{2^0})^0 \cdot (5^{2^1})^1 \cdot (5^{2^2})^1 = (5 \cdot 5^2 \cdot 5^4) = 15625.$$

Si osservi come i fattori tra parentesi possano essere calcolati come quadrati del fattore precedente.

Il costo computazionale del metodo appena illustrato, espresso in numero di moltiplicazioni e quadrati necessari per portare a termine la computazione è uguale a: $\lceil \lg_2 n \rceil$ quadrati + $\frac{1}{2}\lceil \lg_2 n \rceil$ moltiplicazioni (caso medio).

Algorithm 2.2: S&M Right to Left

Input: a, n, $t = \lceil \lg_2 n \rceil$, $n = (n_{t-1}, \ldots, n_1, n_0), n \geq 0$
Output: $c = a^n$

```
1  begin
2      if n = 0 then
3          return 1
4      b ← a
5      if n_0 = 1 then
6          c ← a
7      else
8          c ← 1
9      for i ← 1 to t − 1 do
10         b ← b^2
11         if n_i = 1 then
12             c ← c · b
13     return c
14 end
```

3 Esercizio 2

Si consideri il campo: $\mathbb{F}_{11} \cong \mathbb{Z}/11\mathbb{Z} \cong \mathbb{Z}_{11}$, *si chiede di:*

1. *determinare il numero di generatori del gruppo moltiplicativo;*
2. *esibire il valore di tutti i generatori del campo;*
3. *elencare tutti i sottogruppi di* $\mathbb{F}_{11}^*$ *con la loro cardinalità;*
4. *calcolare l'inverso moltiplicativo di* $x = 7$*;*
5. *descrivere il gruppo additivo del campo* $< \mathbb{F}_{11}, + >$*;*

(1) Determinare il numero di generatori del gruppo moltiplicativo.

$$\mathbb{F}_{11}^* = \{1, 2, \ldots, 10\};\ n = |\mathbb{F}_{11}^*| = 10;$$

$$\text{numero generatori} = \varphi(n) = \varphi(10) = \varphi(2 \cdot 5) = 4$$

o anche

$$\text{numero generatori} = |\{0 < i < n : MCD(i, n) = 1\}| = |\{1, 3, 7, 9\}| = 4.$$

(2) Esibire il valore di tutti i generatori del campo.

Tenendo presente la scomposizione in fattori primi dell'ordine del gruppo

$n = 2 \cdot 5$, assumendo come $g = 2$ come potenziale generatore, si ha:
$2^2 \equiv_{11} 4$, $2^3 \equiv_{11} 8$, $2^4 \equiv_{11} 5$, $2^5 \equiv_{11} 10$, dunque concludiamo che effettivamente $g_0 = g = 2$ è un generatore del gruppo moltiplicativo del campo assegnato. Gli altri 3 generatori sono:

$g_1 = g_0^3 = 2^3 \equiv_{11} 8$;

$g_2 = g_0^7 = 2^7 \equiv_{11} 7$;

$g_3 = g_0^9 = 2^9 \equiv_{11} 6$.

Come verifica, si può provare ad elencare tutti gli elementi del sottogruppo generato da g_3: $< g_3 >= \{g_3 = 6,\ g_3^2 = 3,\ g_3^3 = 7,\ g_3^4 = 9,\ g_3^5 = 10,\ g_3^6 = 5,\ g_3^7 = 8,\ g_3^8 = 4,\ g_3^9 = 2,\ g_3^{10} = 1\}$.

(3) Elencare tutti i sottogruppi di $\mathbb{F}_{11}^*$ con la loro cardinalità;

$$\mathbb{F}_{11}^* = \{1, 2, \ldots, 10\};\ n = |\mathbb{F}_{11}^*| = 10$$

Conoscendo la scomposizione in fattori primi dell'ordine del gruppo: $n = 2 \cdot 5$ possiamo dire che esisteranno soltanto 2 sottogruppi propri H_1, H_2 aventi cardinalità rispettivamente $n_1 = 2$ e $n_2 = 5$.
$H_1 =< h_1 >$ con h_1 elemento di $\mathbb{F}_{11}^*$ di ordine $n_1 = 2$, $h_1 = g_0^{n/n_1} \equiv_{11} 10$;
$H_2 =< h_2 >$ con h_2 elemento di $\mathbb{F}_{11}^*$ di ordine $n_2 = 5$, $h_2 = g_0^{n/n_2} \equiv_{11} 4$;
Tutti i sottogruppi di $\mathbb{F}_{11}^*$ sono dunque:

$H_0 = \{1\},\ n_0 = 1$

$H_1 = \{10, 1\},\ n_1 = 2$

$H_2 = \{4, 5, 9, 3, 1\},\ n_2 = 5$

$\mathbb{F}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\},\ n = 10$

(3) Calcolare l'inverso moltiplicativo di $x = 7$;

Impiegando l'algoritmo di Euclide si trova:

$p = 11, x = 7 \Rightarrow 1 = MCD(7, 4) = 7\xi + 4\eta$ dove $\xi = 2, \eta = -3 \Rightarrow x^{-1} \equiv_{11} \mathbf{8}$

Analogamente, utilizzando il Piccolo di Fermat e l'algoritmo di esponenziazione S&M Left to Right si ha:

$$x^{-1} \equiv_{11} 7^9 \equiv_{11} 7^{1001_2} \equiv_{11} ((7^2)^2)^2 \cdot 7 \equiv_{11} (5^2)^2 \cdot 7 \equiv_{11} 3^2 \cdot 7 \equiv_{11} \mathbf{8}$$

(4) descrivere il gruppo additivo del campo: $< \mathbb{F}_{11}, + >$;

$< \mathbb{F}_{11}, + >= \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $| < \mathbb{F}_{11}, + > | = 11$, è un gruppo ciclico ($\cong \mathbb{Z}_{11}$) in cui tutti gli elementi ($\neq 0$) sono generatori (ordine primo), pertanto non ammette sottogruppi propri.

$$\text{numero generatori} = \varphi(11) = 10$$

Assumendo, ad esempio, $g = 10$ come generatore, si ha:

$$\begin{aligned} &g \equiv_{11} 10,\ 2g \equiv_{11} 9,\ 3g \equiv_{11} 8 \\ &4g \equiv_{11} 7,\ 5g \equiv_{11} 6,\ 6g \equiv_{11} 5, \\ &7g \equiv_{11} 4,\ 8g \equiv_{11} 3,\ 9g \equiv_{11} 2, \\ &10g \equiv_{11} 1,\ 11g \equiv_{11} 0 \end{aligned}$$

4 Esercizio 3

Si consideri il campo: $\mathbb{F}_{13} \cong \mathbb{Z}/13\mathbb{Z} \cong \mathbb{Z}_{13}$, *si chiede di descrivere completamente il gruppo moltiplicativo di tale campo elencando tutti i suoi sottogruppi con le coro cardinalità e i loro generatori nonché le eventuali relazioni tra i sottogruppi trovati.*

Sol: n = 12, numero di sottogruppi = 6;

$n_0 = 1,\ H_0 =< 1 >= \{1\}$;

$n_1 = 4,\ H_1 =< 8 >=< 5 >= \{1, 5, 8, 12\}$;

$n_2 = 2,\ H_2 =< 12 >= \{1, 12\}$;

$n_3 = 3,\ H_3 =< 3 >=< 9 >= \{1, 3, 9\}$;

$n_4 = 6,\ H_4 =< 8 >=< 5 >= \{1, 3, 4, 9, 10, 12\}$;

$n_5 = 12,\ H_5 =< 2 >=< 6 >=< 7 >=< 9 >= \mathbb{F}_{13}^*$;

$H_0 < H_2 < H_1 < H_5; \quad H_0 < H_3 < H_4 < H_5.$

5 Esercizi sui campi $\mathbb{F}_{p^n}$

5.1 Esercizio 1

Dopo aver verificato che il polinomio $f(x) = x^2 - x - 1 \in \mathbb{F}_3[X]$ *è irriducibile, scrivere le tavole di addizione e moltiplicazione del campo:*

$$\mathbb{F}_{3^2} \cong \mathbb{F}_3[X]/ < f(x) > \cong \mathbb{F}_3(\alpha) \text{ con } \alpha \in \mathbb{F}_{3^2} \backslash \mathbb{F}_3 : f(\alpha) = 0.$$

Il polinomio $f(x) \in \mathbb{F}_3[X]$ non ammette radici in $\mathbb{F}_3$ ($f(0) = 2$, $f(1) = 1$, $f(2) = 1$), ed essendo $deg(f(x)) = 2$ si può concludere che è irriducibile.

+	0	1	2	α	α+1	α+2	2α	2α+1	2α+2
0	0	1	2	α	α+1	2+α	2α	2α+1	2α+2
1	1	2	0	α+1	α+2	α	2α+1	2α+2	2α
2	2	0	1	α+2	α	α+1	2α+2	2α	2α+1
α	α	α+1	α+2	2α	2α+1	2α+2	0	1	2
α+1	α+1	α+2	α	2α+1	2α+2	2α	1	2	0
α+2	α+2	α	α+1	2α+2	2α	2α+1	2	0	1
2α	2α	2α+1	2α+2	0	1	2	α	α+1	α+2
2α+1	2α+1	2α+2	2α	1	2	3	α+1	α+2	α
2α+2	2α+2	2α	2α+1	2	0	1	α+2	α	α+1

·	1	2	α	α+1	α+2	2α	2α+1	2α+2
1	1	2	α	α+1	α+2	2α	2α+1	2α+2
2	2	1	2α	2α+2	2α+1	α	α+2	α+1
α	α	2α	α+1	2α+1	1	2α+2	2	α+2
α+1	α+1	2α+2	2α+1	2	α	α+2	2α	1
α+2	α+2	2α+1	1	α	2+2α	2	α+1	2α
2α	2α	α	2α+2	α+2	2	α+1	1	2α+1
2α+1	2α+1	α+2	2	2α	α+1	1	α+2	α
2α+2	2α+2	α+1	α+2	1	2α	2α+1	α	α+1

(1) Determinare: gli elementi primitivi del campo; indicare tutti i possibili polinomi irriducibili e polinomi primitivi che permettono di costruire il campo assegnato.

$$\mathbb{F}_{3^2} \cong \mathbb{F}_3(\alpha) = \theta_0 + \alpha\theta_1 : \theta_0, \theta_1 \in \mathbb{F}_3; \alpha \in \mathbb{F}_{3^2} \backslash \mathbb{F}_3 : f(\alpha) = \alpha^2 - \alpha - 1 = 0.$$

Il numero di polinomi irriducibili di grado 2: $N_2(3) = (3^2 - 3)/2 = 3$; Per trovare i tre polinomi, viste le ridotte dimensioni del campo procediamo nel seguente modo: $f(x) = x^2 + \theta_1 x + \theta_0$; $f(0) \neq 0 \Rightarrow \theta_0 \neq 0 \Leftrightarrow \theta_0 = 1, 2$; dividiamo i due casi:
(1°caso): $f(x) = x^2 + \theta_1 x + 1$, dobbiamo imporre: $f(2) = 2 + 2\theta_1 \neq 0 \Rightarrow \theta_1 \neq 2$; $f(1) = 2 + \theta_1 \neq 0 \Rightarrow \theta_1 \neq 1$; abbiamo così trovato un primo polinomio irriducibile:

$$f_1(x) = x^2 + 1$$

(2°caso): $f(x) = x^2 + \theta_1 x + 2$, dobbiamo imporre: $f(1) = \theta_1 \neq 0$; $f(2) = 6 + 2\theta_1 \neq 0 \Rightarrow \theta_1 \neq 0$; abbiamo così individuato gli altri due polinomi:

$$f_2(x) = x^2 + x + 2$$
$$f_3(x) = x^2 + 2x + 2 = x^2 - x - 1$$

Sia $n = |\mathbb{F}_{3^2}^*|$ la cardinalità del gruppo moltiplicativo, allora il numero di elementi primitivi del campo è: $\varphi(n) = \varphi(8) = 4$; il numero di polinomi primitivi di grado 2 è invece: $M_2(3) = \varphi(8)/2 = 2$. Si sono individuati 3 polinomi irriducibili, due di questi saranno primitivi.
Iniziamo con il trovare un elemento generatore di $\mathbb{F}_{3^2}^*$, usando il polinomio irriducibile $f(x) = x^2 - x - 1$ per fare i calcoli ($f(\alpha) = 0 \Leftrightarrow \alpha^2 = \alpha + 1$).

$$\alpha^1 = \alpha;\ \alpha^2 = \alpha + 1;\ \alpha^3 = 2\alpha + 1;\ \alpha^4 = 2 \Rightarrow \alpha \text{ è primitivo.}$$

(Oss.: i divisori dell'ordine n sono 1,2,4,8.)
Trovato un elemento primitivo $\beta_0 = \alpha$, gli altri elementi primitivi saranno dunque: $\beta_1 = \alpha^3 = 2\alpha + 1$, $\beta_2 = \alpha^5 = 2\alpha$, $\beta_3 = \alpha^7 = \alpha + 2$.
Per calcolare i polinomi primitivi richiesti, applichiamo la definizione e troviamo:

$$g_1(x) = (x - \alpha)(x - \alpha^{3^1}) = x^2 + 2x + 2 = x^2 - x - 1$$
$$g_2(x) = (x - \beta_1)(x - \beta_1^{3^1}) = (x - \alpha^5)(x - \alpha^{15}) = x^2 + x + 2$$

(2) Calcolare l'inverso dei seguenti elementi in $\mathbb{F}_{3^2} \cong \mathbb{F}_3[X]/ < x^2 - x - 1 >$.

$$g(x) = x + 1 \qquad h(x) = 2x$$

Impiegando il Piccolo di Fermat si trova:

$$\begin{aligned}(g(x))^{-1} &= (x+1)^{8-1}\ mod\ f(x) = (x+1)^{111_2}\ mod\ f(x) = \\ &= ((x+1)^2(x+1))^2(x+1)\ mod\ f(x) = \cdots = 2x + 2.\end{aligned}$$

$$\begin{aligned}(h(x))^{-1} &= (2x)^{8-1}\ mod\ f(x) = (2x)^{111_2}\ mod\ f(x) = \\ &= ((2x)^2(2x))^2(2x)\ mod\ f(x) = \cdots = 2x + 1.\end{aligned}$$

Analogamente volendo utilizzare l'algoritmo esteso di Euclide si ha:

$f(x) = x^2 - x - 1, g(x) = x + 1 \Rightarrow 1 = MCD(f(x), g(x)) = f(x)\xi(x) + g(x)\eta(x)$

$$\begin{cases} \underline{u} \leftarrow (x^2 - x - 1, 1, 0); \\ \underline{v} \leftarrow (x + 1, 0, 1); \end{cases}$$

$$\begin{cases} q = \lfloor \frac{x^2 - x - 2}{x+1} \rfloor = x - 2,\ \underline{w} \leftarrow (x^2 - x - 1 - (x+1)(x-2), 1, -(x-2)); \\ \underline{u} \leftarrow (x + 1, 0, 1); \\ \underline{v} \leftarrow (1, 1, -x + 2); \end{cases}$$

$$\begin{cases} q = \lfloor \frac{x+1}{1} \rfloor = x + 1,\ \underline{w} \leftarrow (x + 1 - (x+1), 0 - (x+1), 1 - (-x+2)(x+1)); \\ \underline{u} \leftarrow (1, 1, 2x + 2); \\ \underline{v} \leftarrow (0, 2x + 2, x^2 - x - 1); \end{cases}$$

$$d(x) = 1;\ \xi(x) = 1;\ \eta(x) = 2x + 2 \Rightarrow (g(x))^{-1}\ mod\ f(x) = \mathbf{2x + 2}$$

$$f(x) = x^2 - x - 1, h(x) = 2x \Rightarrow 1 = MCD(f(x), g(x)) = f(x)\xi(x) + g(x)\eta(x)$$

$$\begin{cases} \underline{u} \leftarrow (x^2 - x - 1, 1, 0); \\ \underline{v} \leftarrow (2x, 0, 1); \end{cases}$$

$$\begin{cases} q = \lfloor \frac{x^2 - x - 1}{2x} \rfloor = 2x + 1,\ \underline{w} \leftarrow (-1, 1, x - 1); \\ \underline{u} \leftarrow (2x, 0, 1); \\ \underline{v} \leftarrow (-1, 1, x - 1); \end{cases}$$

$$\begin{cases} q = \lfloor \frac{2x}{-1} \rfloor = x,\ \underline{w} \leftarrow (0, 2x, 2x^2 + x + 1); \\ \underline{u} \leftarrow (-1, 1, x - 1); \\ \underline{v} \leftarrow (0, 2x, 2x^2 + x + 1); \end{cases}$$

$$d(x) = \xi(x)f(x) + \eta(x)h(x), \quad d(x) = -1;\ \xi(x) = 1;\ \eta(x) = x - 1 \Rightarrow$$
$$-1 = (1)f(x) + (x - 1)h(x) \Leftrightarrow$$
$$1 = (-1)f(x) + (2x + 1)h(x) \Rightarrow$$
$$\eta(x) = (h(x))^{-1}\ mod\ f(x) = \mathbf{2x + 1}\ mod\ f(x)$$

5.2 Esercizio 2

Descrivere il campo: $\mathbb{F}_{2^3}$.

Detto f(x) il polinomio irriducibile utilizzato per rappresentare il campo si ha: $\mathbb{F}_{2^3} \cong \mathbb{F}_2(\alpha) = \{\theta_0 + \alpha\theta_1 + \alpha^2\theta_2 : \theta_0, \theta_1, \theta_2 \in \mathbb{F}_2;\ f(\alpha) = 0\}$.

Il numero di elementi generatori del campo $= \varphi(|\mathbb{F}^*_{2^3}|) = \varphi(7) = 6$, quindi tutti gli elementi $\neq 0, 1$ sono generatori.
Il numero di polinomi irriducibili di grado $3 = N_3(2) = (2^3 - 2)/3 = 2$.
Il numero di polinomi primitivi di grado $3 = M_3(2) = \varphi(7)/3 = 2$. risultato scontato, dopo aver osservato che tutti gli elementi del gruppo moltiplicativo sono generatori.
Per esibire i polinomi primitivi/irriducibili, essendo l'estensione del campo uguale a 3, i polinomi per essere irriducibili non devono ammettere come radici 0 o 1, cioè devono avere un numero dispari di coefficienti e un termine noto non nullo:

$$f_1(x) = x^3 + x^2 + 1; \qquad f_2(x) = x^3 + x + 1$$

5.3 Esercizio 3

Descrivere il campo: $\mathbb{F}_{2^5}$.

Il numero di polinomi irriducibili di grado $5 = N_5(2) = (2^5 - 2)/5 = 6$.
Il numero di elementi generatori del campo $= \varphi(|\mathbb{F}_{2^5}^*|) = \varphi(31) = 30$, quindi tutti gli elementi $\neq 0, 1$ sono generatori; il numero di polinomi primitivi di grado 5 coincide con il numero dei polinomi irriducibili.
Per esibire i 6 polinomi primitivi del campo, iniziamo con l'elencare tutti i possibili trinomi e pentanomi a coefficienti in $\mathbb{F}_2$ (Condizione necessaria per l'irriducibilità, è che i polinomi in oggetto non ammettano radici nel campo base $\mathbb{F}_2$):

$$
\begin{aligned}
f_1(x) &= x^5 + x^4 + x^3 + x^2 + 1 \\
f_2(x) &= x^5 + x^4 + x^3 + x + 1 \\
f_3(x) &= x^5 + x^4 + x^2 + x + 1 \\
f_4(x) &= x^5 + x^3 + x^2 + x + 1 \\
f_5(x) &= x^5 + x^4 + 1 \\
f_6(x) &= x^5 + x^3 + 1 \\
f_7(x) &= x^5 + x^2 + 1 \\
f_8(x) &= x^5 + x + 1
\end{aligned}
$$

Applicando il seguente test di irriducibilità, ai polinomi adesso elencati, si trova che $f_5(x)$ e $f_8(x)$ NON sono irriducibili.

Teorema 5.1 (Test di irriducibilità)**.** *Sia $f(x) \in \mathbb{F}_p[X]$ un polinomio monico di grado m, allora $f(x)$ è irriducibilie se e soltanto se:*

$$MCD(f(x), x^{p^h} - x) = cost \quad \forall h \leq [m/2]$$

5.4 Esercizio 4

Considerato il campo $\mathbb{F}_{3^4} \cong \mathbb{F}_3[X]/ < f(x) >$:
(1) indicare il numero di elementi generatori;
(2) il numero di polinomi irriducibili;
(3) il numero dei polinomi primitivi;
(4) verificare che il polinomio $f(x) = x^4 + x - 1 \in \mathbb{F}_3[X]$ è irriducibile.
(5) verificare che l'elemento $\alpha \in \mathbb{F}_{3^4} \backslash \mathbb{F}_3 \ : \ f(\alpha) = 0$ è un elemento primitivo del campo.
(6) Sapendo che $f(x) = (x - \alpha)(x - \alpha^3)(x - \alpha^9)(x - \alpha^{27}) = x^4 + x - 1$ è primitivo, determinare un altro polinomio primitivo del campo, giustificando il procedimento. (Sugg.: considerare l'elemento $\beta = \alpha^7$)

Sol: $n = 80$*, numero generatori* $= 32$*; numero pol. irriducibili* $= 18$*;*
numero pol. primitivi = 8.
Condizione necessaria perchè il polinomio $f(x) = x^4 + x - 1$ *sia irriducibile:* $f(0) = 2 \neq 0$*,* $f(1) = 1 \neq 0$*,* $f(2) = 2 \neq 0$*, quindi il polinomio dato non ammette fattori di primo grado a coefficienti in* $\mathbb{F}_3$*, quello che rimane da verificare, essendo il polinomio di quarto grado è che non ammetta fattori di secondo grado:*

$$(x^4 + x + 1) = (x^2 + ax + b)(x^2 + cx + d) \quad a, b, c, d \in \mathbb{F}_3$$

eseguendo i prodotti si imposta il sistema:

$$\begin{cases} a + c = 0 \\ d + ac + b = 0 \\ ad + bc = 0 \\ bd = -1 \end{cases}$$

osservando la prima e l'ultima equazione si vede come i valori possibili per $a,b,c,d \in \mathbb{F}_3$ *siano:* $a = 1$*,* $d = -1$ *oppure* $a = -1$*,* $d = 1$*; mentre* $a = 1$*,* $c = -1$ *oppure* $a = -1$*,* $c = 1$*; e per ognuna di queste combinazioni di valori si vede come la seconda equazione non sia mai soddisfatta. Si può dunque concludere che il polinomio* $f(x) = x^4 + x - 1 \in \mathbb{F}_3[X]$ *è irriducibile.*
Fissando per i calcoli il polinomio $f(x) = x^4 + x - 1$*, un altro polinomio primitivo del campo è:* $g(x) = x^4 + x^3 - x^2 - x - 1$
Nota:

$$\begin{aligned} &\alpha^4 = -\alpha + 1; \\ &\alpha^5 = -\alpha^2 + \alpha \\ &\alpha^6 = -\alpha^3 + \alpha^2; \\ &\alpha^7 = \alpha^3 + \alpha - 1 \\ &\alpha^8 = \alpha^2 + \alpha + 1; \\ &\alpha^9 = \alpha^3 + \alpha^2 + \alpha \\ &\alpha^{10} = \alpha^3 + \alpha^2 - \alpha + 1 \\ &\alpha^{16} = -\alpha^3 + \alpha - 1 \\ &\alpha^{20} = -\alpha^3 - \alpha^2 + \alpha \\ &\alpha^{40} = 2; \end{aligned}$$

Sia G un gruppo abeliano finito $(G, \circ)$
Dati due elementi $x, y \in G$, il log discreto di y in base x è dato dall'intero m tale che

$$x^m = y \; ; \; m = \log_x^D(y) \; ; \; m \in \{0, 1, \dots, |G|-1\}$$

In un gruppo abeliano qualsiasi, non è vero in generale che esista il log discreto di un elemento in una base arbitraria ma se $\log^D$ esiste allora è unico.

Esempio: $G = \mathbb{F}_7^*$ è un gruppo ciclico finito ($\Rightarrow$ quindi abeliano)
se si prende $x=2$ e $y=3$ allora

$m = \log_x^D y = \log_2^D 3$ ~~NON~~ ESISTE

infatti: $2^1 \equiv_7 2$, $2^2 \equiv_7 4$, $2^3 \equiv_7 1$

Se si considera invece $x=5$, $y=3$ allora

$m = \log_x^D y = \log_5^D 3 = 5$;

infatti: $5^1 \equiv 5$; $5^2 \equiv 4$; $5^3 \equiv 6$; $5^4 \equiv 2$; $5^6 \equiv 3$.

↳ posso già dire che $\log_5^D 3$ esiste x'è 5 è un generatore (i divisori di $|G|=6$ sono solo 1, 2, 3, 6)

Quindi, se si considera un gruppo ciclico finito e si assume come base dei logaritmi un elemento generatore, il logaritmo discreto di ogni elemento del gruppo esiste sempre.

Avendo a disposizione un gruppo $(G, \circ)$ finito, non necessariamente ciclico, ci si può ricondurre ad un sottogruppo appropriato (ciclico).

Teorema: In un gruppo abeliano finito di ordine $|G| = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$, esiste almeno un sottogruppo di ordine m se e soltanto se $m \mid |G|$.

Se conoscessimo la fattorizzazione di $|G| = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$ si può facilmente estrarre un generatore di un sottogruppo ciclico di ordine p_1:

(1) Si estrae un elemento casuale di G, diciamo g.

(2) Si calcola $\alpha = g^c$ dove l'intero c è tale che $|G| = p_1 \cdot c$.
Se $\alpha \neq e$ (elemento neutro del gruppo) allora α è un generatore del sottogruppo ciclico di ordine p_1 da esso generato: $\langle g \rangle$.

Esempio: $G = (\mathbb{Z}_{22}, +)$; $|G| = 22 = 2 \cdot 11$; Vogliamo lavorare nel sottogruppo $H < G$ di ordine $p_1 = 11$; quindi $c = 2$

(1) $g \leftarrow$ random $(22) = 5$

(2) $\alpha = c \cdot g = 2 \cdot 5 = 10 \bmod 22 \neq 0$ (elemento neutro di G)
$H = \langle \alpha \rangle = \langle 10 \rangle$.

Ovviamente, nel gruppo ciclico finito $G = \langle g \rangle$, $n = |G|$, valgono le proprietà formali dei logaritmi:

$a, b \in G$; $s \in \mathbb{Z}$

$\rightarrow \log_g^D(ab) \equiv \log_g^D a + \log_g^D b \pmod n$

$\rightarrow \log_g^D(a^s) \equiv s \log_g^D a \pmod n$.

- Si può dunque formalizzare il così-detto GDLP (Generalized Discrete Logarithm Problem)

Assegnato un gruppo ciclico G con un elemento primitivo g. Per ogni $\alpha \in G$, Trovare l'unico esponente positivo m: $g^m = \alpha$ con $0 \leq m \leq |G|$.

In crittografia è importante selezionare gruppi ciclici G, in cui il GDLP sia un problema "computazionalmente difficile"

Esempio: se $G = (\mathbb{Z}_{19}, +)$; $|G| = 19$ (ciclico); $g = 2$;

se $x = 15$; $\log_g^D x = \log_2^D 15 = m$

m $g^m = x \Leftrightarrow 2 \cdot m = 15 \bmod 19 \Rightarrow$

con l'algoritmo di Euclide si trova 2^{-1} e dunque

$m = 2^{-1} \cdot 15 \bmod 19 \equiv 10 \cdot 15 \equiv 17 \bmod 19$.

La complessità comput. dell'Algo di Euclide (ottimizzato) è $O(2 \log_2 |G|)$ dunque visto che anche la moltiplicazione è $O(\log_2 |G|) \Rightarrow O(\log_2 |G|)$.

Quindi non si utilizzeranno mai gruppi additivi $G_1 = (\mathbb{Z}_p, +)$ ma, tipicamente, il gruppo moltiplicativo di un campo finito $\mathbb{F}_q^*$.

Esempio: $G_1 = (\mathbb{F}_{2^5}^*, \cdot)$ con $f(x) = x^5 + x^2 + 1$, $n = |G_1| = 31$.
$g = (0010)$ elemento generatore.

$G_1 = (\mathbb{F}_{31}^*, \cdot)$ con $n = |G_1| = 30$ e generatore $g = 3$.

Non esistono algoritmi per l'estrazione dei log discreti in gruppi moltiplicativi che ammettano una complessità computazionale polinomiale;
Il meglio che si può ottenere è una complessità del tipo

$$O(\exp(c(\lg q)^{v}(\lg \lg q)^{1-v}))$$

↓ lunghezza in bit degli elementi del gruppo.

Dunque, prendendo gruppi "sufficientemente" ampi si può garantire il livello di sicurezza desiderato. Quando il DLP con $G_1 = \mathbb{F}_q^*$ è assunto come paradigma per la costruzione di schemi crittografici a chiave pubblica.

CRITTOGRAFIA A CHIAVE PUBBLICA

→ Coppia di chiavi (K_{pub}, K_{priv}) usate rispettivamente in cifratura e decifrazione.

→ Crittosistema: $(\mathcal{A}, \mathcal{M}, \mathcal{C}, \mathcal{K}, \{E_e : e \in K\}, \{D_d : d \in K\})$

$\mathcal{A}$: insieme finito, detto alfabeto di definizione dei messaggi; es.: $\mathcal{A} = \{0,1\}$ (stringhe di bit)

$\mathcal{M}$: insieme dei messaggi in chiaro; ogni elemento è rappresentato come una stringa di simboli dell'alfabeto di lunghezza variabile arbitrariamente prefissata.

$\mathcal{C}$: insieme dei messaggi cifrati; anch'esso costituito da elementi composti come stringhe dell'alfabeto; ovviamente $|\mathcal{M}| = |\mathcal{C}|$.

$\mathcal{K}$: insieme delle chiavi. Per ogni elemento $e \in \mathcal{K}$ è possibile individuare una biiezione

$$E_e : \mathcal{M} \longrightarrow \mathcal{C}$$

detta FUNZIONE DI CIFRATURA CON "e" = CHIAVE PUBBLICA

Il concetto fondamentale è che la funzione inversa di questa biiezione non deve essere "facilmente" invertibile a meno che non si conosca un parametro $d \in \mathcal{K}$ come unica possibilità di individuare la FUNZIONE DI DECIFRAZIONE (meglio calcolare)

$$D_d : \mathcal{C} \longrightarrow \mathcal{M} \; ; \text{"d"} = \text{CHIAVE PRIVATA}$$

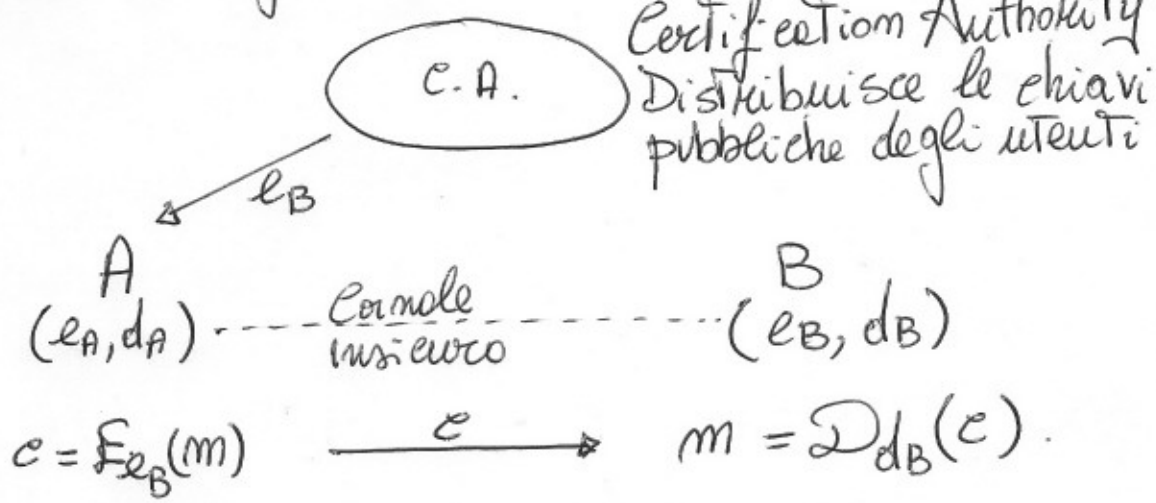
Altra proprietà fondamentale che deve essere garantita dalla scelta dell'insieme K e che siano rispettate le seguenti relazioni:

$$\mathcal{D}_d(\mathcal{E}_e(m)) = m \in \mathcal{M};$$

$$\mathcal{E}_e(\mathcal{D}_d(m)) = m \in \mathcal{M};$$

- La costruzione delle funzioni $\mathcal{E}_e(\cdot)$, $\mathcal{D}_d(\cdot)$ è PUBBLICAMENTE NOTA:

Schema di cifratura:



Schema di firma:

m: messaggio da inviare e firmare.

$h(m)$: "digest" di m; $h(\cdot)$ FUNZIONE DI HASH, pubblicamente nota.

$$c = \mathcal{E}_{e_B}\left(m \,\|\, \mathcal{D}_{d_A}(h(m))\right)$$

"CONCATENAZIONE"

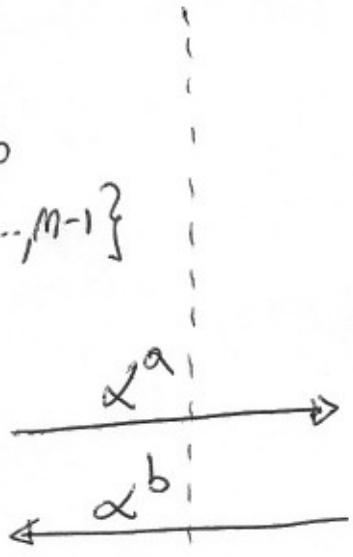
c

1. $\mathcal{D}_{d_B}(c) = m \,\|\, \mathcal{D}_{d_A}(h(m))$
2. Calcola $h(m)$
3. check: $\mathcal{E}_{e_A}(\mathcal{D}_{d_A}(h(m))) \stackrel{?}{=} h(m)$

Scambio di chiavi Diffie – Hellmann

Sia G un gruppo finito abeliano, $n = |G|$, $\alpha \in G$ un suo generatore.

Si vuole stabilire un <u>segreto comune</u> tra due interlocutori A, B con uno scambio di messaggi in pubblico (su canale insicuro)



$$k_{AB} = k_{BA}.$$

Esempio: $G_1 = \mathbb{F}_{5^2}^*$; $f(x) = x^2 + x + 2$;

$\mathbb{F}_{5^2} \cong \mathbb{F}_5(\alpha) = \{\vartheta_0 + \alpha\vartheta_1 ; \vartheta_0, \vartheta_1 \in \mathbb{F}_5 ; \alpha^2 = -\alpha - 2 ; \alpha \in \mathbb{F}_{5^2}^* \setminus \mathbb{F}_5\}$

$n = |G_1| = 24$;

α È UN ELEMENTO GENERATORE (VERIFICARE!)

A

1. $a = \text{random}(n) = 13$
2. $\alpha^a = \alpha^{13} = \alpha^{1101_2} =$
   $= ((\alpha^2 \cdot \alpha)^2)^2 \cdot \alpha =$
   $= \dots = 4\alpha$

B

1. $b = \text{random}(24) = 3$
2. $\alpha^b = \alpha^3 = -\alpha^2 - 2\alpha \equiv$
   $\equiv 4\alpha + 2$

4α →

← $4\alpha + 2$

3. $k_{AB} = (4\alpha+2)^{13} = (4\alpha+2)^{1101_2} =$
   $\equiv (-\alpha+2)^{1101_2} \equiv$
   $\equiv \dots \equiv \alpha + 3$

3. $k_{BA} = (4\alpha)^3 \equiv (-\alpha)^2(-\alpha) \equiv$
   $\equiv (-\alpha-2)(-\alpha) \equiv$
   $\equiv \alpha^2 + 2\alpha \equiv \alpha - 2 \equiv$
   $\equiv \alpha + 3.$

CRITTOSISTEMA DI EL-GAMAL

Definisce una funzione di cifratura/decifrazione basato su GDLP assumendo $G = \mathbb{F}_q^*$; $q = p^n$.

Set-up:

1. Si fissa $G = \mathbb{F}_q^*$; $n = |G| = q-1$; $\alpha \in G$ GENERATORE
2. Si sceglie (Alice) un numero casuale $b \in \mathbb{Z}$ tale che $0 < b < n$, definendo
$K_{pub,B} = (\alpha, \beta = \alpha^b)$; $K_{priv,B} = (b)$

Scenario: A $\xleftarrow{K_{pub,B}}$ (C.A.)

B
$K_{pub,B} = (\alpha, \beta)$.
$K_{priv,B} = (b)$.

1. $m \in G$ MESSAGGIO IN CHIARO
$\ell = \text{random}(n) \in \mathbb{Z}_n$
2. Calcola
$\begin{cases} \gamma \equiv (\alpha)^\ell \\ \delta \equiv \beta^\ell \cdot m \equiv (\alpha^b)^\ell m \end{cases}$
3. Invia il Testo cifrato
$e = \langle \gamma, \delta \rangle$ $\xrightarrow{\quad e \quad}$

1. Usando la propria chiave privata $K_{priv,B} = b$; Calcola
$\gamma^{n-b} \cdot \delta \equiv \gamma^{-b} \cdot \delta \equiv m$.

VERIFICA: $\gamma^{n-b} \cdot \delta \equiv \gamma^{-b} \cdot \delta \equiv \alpha^{-b\ell} \delta \equiv \alpha^{-b\ell} \alpha^{b\ell} m \equiv m$ ∎

Esempio: $G = \mathbb{F}_{31}^*$; $n = 30$; $\alpha = 3$ generatore

A

1. $m = 12 \in G$;
2. $l = \text{rand}(30) = 3$;
3. $e = E_{K_{pub,B}}(m) = \langle \gamma, \delta \rangle$

$\gamma \equiv \alpha^l \equiv 3^3 \equiv 27 \bmod 31.$

$\delta \equiv \beta^l m \equiv_{31} 26^3 \cdot 12 \equiv_{31} 19.$

B

$K_{pub,B} = (\alpha = 3; \beta = \alpha^b \equiv 26 \bmod 31)$;

$K_{priv,B} = (b = 5)$;

$e = \langle 27, 19 \rangle \rightarrow$

$m = \gamma^{-b}\delta \equiv 27^{-5} \cdot 19 \equiv$

$\equiv 27^{25} \cdot 19 \equiv$

$\equiv (((27^2 \cdot 27)^2)^2) \cdot 27 \cdot 19 \equiv$

$\equiv 30 \cdot 19 \equiv 12 \bmod 31.$

Sicurezza del Crittosistema di El-Gamal.

Chiunque osservi il canale conosce:

$$m, \alpha, \alpha^b, \gamma = \alpha^l, \delta = \beta^l m = m\alpha^{bl}$$

- Si può estrarre $b = \log^D_\alpha \alpha^b$ e ricavare $m = \delta(\gamma^{-1})^b$;
  oppure $l = \log^D_\alpha \gamma$ e ricavare $m = [(\alpha^b)^{-1}]^l \cdot \delta$;
  ma ciò richiederebbe la soluzione di un Log. DISCRETO.
- Non esiste allo stato attuale, nessun metodo conosciuto per calcolare α^{bl} noti α^b e α^l; altrimenti, se ciò non fosse si avrebbe subito $m = \delta(\alpha^{bl})^{-1}$
  Ovviamente se si potesse risolvere il Log. DISCRETO Anche questo Problema sarebbe banalmente risolto.

EL-GAMAL SIGNATURE SCHEME

- Sia dato un insieme di messaggi $\mathcal{M}$ su un alfabeto binario $A=\{0,1\}$.
- Dato, un gruppo ciclico finito G_1 di ordine n, con generatore $\alpha \in G_1$; si assume che anche gli elementi di G_1 siano rappresentabili univocamente come stringhe binarie.
- Si assume la definizione di una funzione di Hash: $h: \{0,1\}^* \longrightarrow \mathbb{Z}_n$
- Le chiavi in possesso di ogni utente sono definite come:

$$k_{pub} = (\alpha, \beta = \alpha^b) \; ; \; k_{priv} = (b) \; ; \; b \in \mathbb{Z}_n$$

- Lo scenario di funzionamento prevede che un utente B firmi un messaggio m usando la propria chiave privata $k_{priv,B}$. L'interlocutore A, avrà a disposizione un algoritmo di verifica della firma con il quale decidere se accettare o rifiutare la firma.

$$B \xrightarrow[k_{PRIV,B}]{<c = Sig(m) \; ; \; m>} A$$

$$Ver_{k_{pub,B}}(c, m) \stackrel{?}{=} \begin{cases} \text{TRUE} \\ \text{FALSE} \end{cases}$$

FIRMA:

B: $k_{PUB,B} = (\alpha, \beta = \alpha^b)$; $k_{PRIV,B} = (b)$, $b \in \mathbb{Z}_n$

1. $l = \text{random}(n) \in \mathbb{Z}_n^* \iff l = \text{random}(n) \in \mathbb{Z}_n$ TALE CHE $MCD(l, n) = 1$.
2. CALCOLA: $\gamma = \alpha^l$ (OPERAZIONI IN G)
3. CALCOLA $l^{-1} \bmod n$ (OPERAZIONI IN $\mathbb{Z}_n$)
4. CALCOLA $h(m), h(\gamma) \in \mathbb{Z}_n$
5. CALCOLA $\delta = l^{-1} \cdot (h(m) - b\,h(\gamma)) \bmod n$ (OPERAZIONI IN $\mathbb{Z}_n$)
6. INVIO DI: $S = Sig_{k_{PRIV,B}}(m) = \langle m, \gamma, \delta \rangle$

VERIFICA:

INPUT: $k_{PUB,B} = (\alpha, \beta = \alpha^b)$; $S = \langle m, \gamma, \delta \rangle$

1. CALCOLA $h(m), h(\gamma)$
2. VERIFICA $Ver_{k_{pub,B}}(m, \gamma, \delta) \triangleq \beta^{h(\gamma)} * \gamma^{\delta} = \begin{cases} \alpha^{h(m)}, \text{ ACCETTA} \\ \neq \alpha^{h(m)}, \text{ RIFIUTA} \end{cases}$

OSSERVAZIONE: $\beta^{h(\gamma)} * \gamma^{\delta} \equiv \alpha^{b h(\gamma)} \alpha^{l\delta} \equiv \alpha^{b h(\gamma)} \alpha^{h(m) - b h(\gamma)} \equiv \alpha^{h(m)}$ ∎

Esempio: $G = \mathbb{F}_{25}^*$; $f(x) = x^5 + x^2 + 1$ irriducibile in $\mathbb{F}_2[X]$; $n = |G| = 31$;

$\alpha = (00010)$ generatore di G; Rappresentazione: $\mathbb{F}_{25} \cong \mathbb{F}_2(\alpha)$ con $f(\alpha) = 0$;

$b = 19$; $m = 10000_2$;

$h : \{0,1\}^* \to \mathbb{Z}_n$ del valore decimale della sequenza binaria modulo n

$h(m) = 16 \bmod 31$.

$k_{pub} = (\alpha, \beta = \alpha^b)$; $\beta = \alpha^{19} = \alpha^{10011_2} = \alpha^4 + \alpha$;

Nota: $\alpha^5 = \alpha^2 + 1$
$\alpha^6 = \alpha^3 + \alpha$
$\alpha^7 = \alpha^4 + \alpha^2$
$\alpha^8 = \alpha^3 + \alpha^2 + 1$

Sign: • $l = \text{random}(31) \in \mathbb{Z}_{31}^* = 24$;

• $\gamma = \alpha^l = \alpha^{24} = \alpha^{11000_2} \equiv (((\alpha^2 \cdot \alpha)^2)^2)^2 \equiv ((\alpha^3 + \alpha)^2)^2 \equiv (\alpha^3 + \alpha^2 + \alpha)^2 \equiv \alpha^4 + \alpha^3 + \alpha^2 + \alpha$;

• $l^{-1} \bmod n = 24^{-1} \bmod 31 \equiv 24^{\varphi(31)-1} \bmod 31 \equiv 24^{29} \bmod 31 \equiv$

$\equiv 24^{11101_2} \equiv (((24^2 \cdot 24)^2 \cdot 24)^2)^2 \cdot 24 \equiv ((29^2 \cdot 24)^2)^2 \cdot 24 \equiv 3^4 \cdot 24 \equiv 19 \cdot 24 \equiv$

$\equiv 22 \bmod 31$.

• $h(m) = 16 \bmod 31$; $h(\gamma) = h(11110) = 30 \bmod 31$;

- $\delta = \ell^{-1}(h(m) - b\,h(\gamma)) = 22(16 - 19\cdot 30) \bmod 31 = 26 \bmod 31.$
- $\text{Smuia} <m, \gamma, \delta> = <\{10000\}, \alpha^4+\alpha^3+\alpha^2+\alpha, 26>$

Verify: $\text{Ver}_{k_{pub}}(m,\gamma,\delta) = \beta^{h(\gamma)}\gamma^{\delta} \stackrel{?}{=} \alpha^{h(m)}$; $h(m) = h(10000) = 16$; $h(\gamma) = h(11110) = 30$;

- $\beta^{h(\gamma)} = (\alpha^2+\alpha)^{30} = (\alpha^2+\alpha)^{11110_2} = (((((\alpha^2+\alpha)^2(\alpha^2+\alpha))^2(\alpha^2+\alpha))^2(\alpha^2+\alpha))^2 =$
  $= (((\alpha^6+\alpha^5+\alpha^4+\alpha^3)^2(\alpha^2+\alpha))^2(\alpha^2+\alpha))^2 =$
  $= (((\alpha^3+\alpha+\alpha^2+1+\alpha^4+\alpha^3)^2(\alpha^2+\alpha))^2(\alpha^2+\alpha))^2 =$
  $= (((\alpha^8+\alpha^4+\alpha^2+1)(\alpha^2+\alpha))^2(\alpha^2+\alpha))^2 = (((\alpha^3+\alpha^2+1+\alpha^4+\alpha^2+1)(\alpha^2+\alpha))^2(\alpha^2+\alpha))^2 =$
  $= ((\alpha^5+\alpha^4+\alpha^6+\alpha^5)^2(\alpha^2+\alpha))^2 = ((\alpha^4+\alpha^3+\alpha)^2(\alpha^2+\alpha))^2 = (((\alpha^3+\alpha^2+1+\alpha^3+\alpha+\alpha^2)(\alpha^2+\alpha))^2 =$
  $= ((\alpha+1)(\alpha^2+\alpha))^2 = (\alpha^3+\alpha^2+\alpha^2+\alpha)^2 = \alpha^6+\alpha^2 = \alpha^3+\alpha^2+\alpha$ ∎

  $\beta^{h(\gamma)} = \alpha^3+\alpha^2+\alpha.$

- $\gamma^{\delta} = (\alpha^4+\alpha^3+\alpha^2+\alpha)^{26} = (((-)^2\cdot(-)^4\cdot(-))^2 = (((\alpha^3+\alpha^2+1+\alpha^3+\alpha+\alpha^4+\alpha^2)(-)^4(-))^2 =$
  $= (\underbrace{(\alpha^8+\alpha^7+\alpha^6+\alpha^5+\alpha^5+\alpha^4+\alpha^3+\alpha^2+\alpha^4+\alpha^3+\alpha^2+\alpha)^4}_{\alpha^3+\alpha^2+1+\alpha^4+\alpha^2+\alpha^5+\alpha+\alpha}\cdot(-))^2 = ((\alpha^3+\alpha^2)^2(-))^2 = ((\alpha^4+\alpha^3+\alpha)(-))^2$
  $= (\alpha^8+\alpha^4+\alpha^6+\alpha^5+\alpha^4+\alpha^6+\alpha^5+\alpha^4+\alpha^5+\alpha^4+\alpha^3+\alpha^2)^2 = (\alpha^3+\alpha^2+1+\alpha^2+1+\alpha^3+\alpha^2)^2 = \alpha^4$ ∎

- $\gamma^{\delta} = (\alpha^4+\alpha^3+\alpha^2+\alpha)^{26} = \alpha^4$;
- $\alpha^{h(m)} = \alpha^{16} = (\alpha^8)^2 = (\alpha^3+\alpha^2+1)^2 = \alpha^6+\alpha^4+1 = \alpha^3+\alpha+\alpha^4+1 \Rightarrow$

$$= \alpha^4+\alpha^3+\alpha+1;$$

$$\beta^{h(\gamma)} \cdot \gamma^{\delta} = (\alpha^3+\alpha^2+\alpha)\cdot(\alpha^4) = \alpha^7+\alpha^6+\alpha^5 = \alpha^4+\cancel{\alpha^2}+\alpha^3+\alpha+\cancel{\alpha^2}+1 =$$

$$= \alpha^4+\alpha^3+\alpha+1;$$

$$\boxed{\beta^{h(\gamma)} \cdot \gamma^{\delta} = \alpha^{h(m)}} \checkmark$$

ESEMPIO: $G = \mathbb{Z}_p^*$; $p = 11$; $n = |G| = 10$; $\alpha = 2$ GENERATORE

$m = \{1001\}$ MESSAGGIO IN CHIARO (IN BINARIO)

$b = 4$ CHIAVE PRIVATA. $k_{PRIV,B} = (b) = 4$; $k_{PUB,B} = (\alpha = 2, \beta = \alpha^b = 5)$

$h : \{0,1\}^* \longrightarrow \mathbb{Z}_n$ $\Leftrightarrow$ VALORE DECIMALE MODULO n DELLA SEQUENZA BINARIA.

FIRMA: • $\ell = \text{random}_n(10) \in \mathbb{Z}_n^* = 3$;

• $\gamma = \alpha^\ell \equiv 2^3 \equiv 8 \bmod 11$.

• $\ell^{-1} \bmod n = 3^{-1} \bmod 10 = 3^{\varphi(10)-1} \bmod 10 = 3^3 \bmod 10 = 7 \bmod 10$;

• $h(m) = h(\{1001\}) = 9 \bmod 10$; $h(\gamma) = h(\{1000\}) = 8 \bmod 10$;

• $\delta = \ell^{-1}(h(m) - b\,h(\gamma)) \equiv 7(9 - 4 \cdot 8) \bmod 10 \equiv 9 \bmod 10$;

• INVIA: $\langle m, \gamma, \delta \rangle = \langle \{1001\}, 8, 9 \rangle$

VERIFICA: • $h(m) = h(\{1001\}) = 9 \bmod 10$; $h(\gamma) = h(8) = h(\{1000\}) = 8 \bmod 10$;

• $\beta^{h(\gamma)} \equiv 5^8 \bmod 11 = 5^{1000_2} \bmod 11 = ((25)^2)^2 \equiv 4 \bmod 11$;

$\gamma^\delta \equiv 8^9 \bmod 11 = 8^{1001_2} \bmod 11 \equiv_{11} ((8^2)^2)^2) \cdot 8 \equiv_{11} (9^2)^2 \cdot 8 \equiv_{11} 4^2 \cdot 8 \equiv_{11} 7$;

$\alpha^{h(m)} \equiv 2^9 \bmod 11 \equiv 6 \bmod 11$. ; $\beta^{h(\gamma)} \gamma^\delta \equiv_{11} 4 \cdot 7 \equiv_{11} 6 \equiv \alpha^{h(m)}$ ∎

RSA

Il più popolare sistema crittografico a chiave pubblica, basato sul problema della fattorizzazione di interi (non ammette algoritmi risolutori a complessità polinomiale, ma soltanto sub-esponenziale come per i DLog)

Scenario: B invia un msg cifrato a A dopo aver recuperato la $k_{pub,A}$.

(A)

Set-up

1. Sceglie due primi $p, q \approx 2^{512}$
2. calcola $n_A = p \cdot q$;
3. $e_A = \text{random}(\varphi(n_A)) \in \mathbb{Z}^*_{\varphi(n_A)}$
   $d_A = e_A^{-1} \bmod \varphi(n_A)$
4. $k_{pub,A} = (n_A, e_A)$;
   $k_{priv,A} = (p, q, \varphi(n_A), d_A)$;

(B)

$k_{pub,A}$ →

Encryption

1. $m \in \mathbb{Z}_{n_A}$
2. $c_A = E_{k_{pub,A}}(m) = m^{e_A} \bmod n_A$

← c_A

Decryption

1. $D_{k_{priv,A}}(c_A) = c_A^{d_A} \bmod n_A$
   $= m$.

Occorre dimostrare che

$$D_{K_{priv,A}}\left(E_{K_{pub,A}}(m)\right) = m \in \mathbb{Z}_{n_A}$$

cioè: $c_A^{d_A} \bmod n_A = m^{e_A d_A} \bmod n_A = m \bmod n_A$?

$e_A d_A \equiv 1 \bmod \varphi(n_A)$ quindi $e_A d_A = 1 + t\varphi(n_A)$; $t \in \mathbb{Z}$

(*) $m^{e_A d_A} \bmod n_A = m\left(m^{\varphi(n_A)}\right)^t \bmod n_A$

2 casi: (i) $MCD(m, n_A) = 1$ quindi vale il Teorema di Eulero e $\left(m^{\varphi(n_A)}\right)^t \bmod n_A \equiv 1 \bmod n_A$ ∎

(ii) $MCD(m, n_A) = d \neq 1$ ma allora $MCD(m, p\cdot q) \neq 1$ sarà $m = r\cdot p$ con $r < q$ oppure $m = s\cdot q$ con $s < p$

supponiamo $m = r\cdot p$, allora

$$m^{\varphi(n_A)} = m^{(p-1)(q-1)} = \left(m^{\varphi(q)}\right)^{(p-1)} \equiv 1 \bmod q$$

dunque $\left(m^{\varphi(n_A)}\right)^t \equiv 1 \bmod q \iff$

$\left(m^{\varphi(n_A)}\right)^t = 1 + cq$; $c \in \mathbb{Z}$

TORNANDO ALLA (*) SI HA:

$$m^{e_A d_A} \bmod n_A = m(1+cq) = m + rpcq \bmod n_A = m \bmod n_A$$ ∎

Attacchi al crittosistema

- Forza bruta: Dato $c_A = m_A^{e_A} \bmod n_A$, provare tutte le possibili chiavi $d \in [1, \varphi(n_A)]$ per trovare quella t.c. $c_A^{d_A} = m$
  ma $\varphi(n_A) \approx n_A \approx 2^{512} \longrightarrow$ impossibile!

- Trovare $\varphi(n_A)$: Tale numero deve essere mantenuto segreto come p e q.
  Dati $n_A, e_A, c_A = m_A^{e_A} \bmod n_A$, lo scopo è trovare $\varphi(n_A)$ e quindi ricavare $d_A = e_A^{-1} \bmod \varphi(n_A)$ con algoritmo di Euclide.

Calcolare $\varphi(n_A)$ SENZA conoscere la fattorizzazione di n_A è un problema difficile quanto la fattorizzazione di n_A.

$$CALC_{\varphi(n_A)} \leq_P FATT_{n_A}$$

Infatti se $n_A = \prod_{i=1}^{k} p_i^{e_i}$ allora $\varphi(n_A) = \prod_{i=1}^{k} (p_i^{e_i} - p_i^{e_i - 1})$

$$FATT_{n_A} \leq_P CALC_{\varphi(n_A)}$$

Infatti: Nota $\varphi(n_A)$ si ha: $\varphi(n_A) = (p-1)(q-1) = n - (p+q) + 1$
si imposta l'eq. $X^2 + [\varphi(n_A) - n - 1]X + n_A = 0$
e la si risolve, trovando p e q.

Aspetti Computazionali

(i) Per determinare p, q primi $\sim 2^{512}$
si estrae un numero casuale e
si applica un Test di Primalità.
Test di Miller-Rabin.

(ii) Per estrarre $e_A \in \mathbb{Z}^*_{\varphi(M_A)}$

→ Calcola $\varphi(M_A)$

→ do { $e_A \leftarrow$ random $(\varphi(M_A))$;
Applica l'algor. esteso di euclide
$d = MCD(\varphi(M_A), e_A) = \gamma\varphi(M_A) + \delta e_A$
} while $d \neq 1$

$d_A \leftarrow \delta$.

(iii) Algoritmi di Fattorizzazione

→ Quadratic Sieve (QS): $O\left(e^{(1+o(1))\sqrt{\ln n \ln\ln n}}\right)$

→ Elliptic Curve: Usa il fattore + Piccolo tra p e q $\quad O\left(e^{(1+o(1))\sqrt{2(\ln p)(\ln\ln p)}}\right)$

→ Number Field Sieve: $O(L_n(1/3, 1.92))$

$L_n(v, c) = e^{c(\ln n)^v (\ln\ln n)^{1-v}}$.

- L'esponente di cifratura e_A nell'RSA può essere scelto con un basso peso di hamming senza intaccare la sicurezza del crittosistema, ma accelerando il calcolo (vedi Tecniche di exp veloce)
- L'esponente di decifrazione $d_A = e_A^{-1} \bmod \varphi(n_A)$ similmente non avrà un basso peso di hamming, per accelerare quindi il calcolo di $c_A^{d_A} \bmod n_A$ si può applicare il Teorema cinese dei resti per effettuare i calcoli modulo p e modulo q, anziché modulo n_A.

$$\begin{cases} c_A^{d_A} \bmod p = X \\ c_A^{d_A} \bmod q = X \end{cases} \xRightarrow{CRT}$$ fornisce l'unica soluzione $X \bmod (p \cdot q) = X \bmod n_A$ che soddisfa il sistema cioè: $X = c_A^{d_A} \bmod n_A$.

TEOREMA CINESE RESTO

Dato un intero X, $N = \prod_{i=1}^{k}$ con $(n_i, n_j) = 1 \ \forall i, j : i \neq j$

la corrispondenza

$$X \longleftrightarrow \{x_1, x_2, \dots, x_k\} \text{ con } x_i = X \bmod n_i \ \forall i$$

è biunivoca.

Traccia di Dim:

- che Dato X, $\exists !$ tupla $\{x_1, \dots, x_k\}$ con $x_i = X \bmod n_i$ è evidente.

- Dimostriamo, in maniera costruttiva, che data la tupla $\{x_1, \dots, x_k\}$ a questa corrisponde un'unico intero $X \pmod N$

$$M_i \triangleq N / n_i \ ; \quad M_i^{-1} \cdot M_i \equiv 1 \bmod n_i$$

Oss: $M_i M_i^{-1} \equiv 1 \bmod n_i$

$M_i M_j^{-1} \equiv 0 \bmod n_j$

$$X \triangleq \left(\sum_{i=1}^{k} x_i M_i M_i^{-1} \right) \bmod N.$$

→ Nelle implementazioni pratiche di RSA per applicare il Teorema Cinese del Resto si utilizza il così-detto GARNER'S ALGORITHM che non impiega alcuna inversione ~~particolare~~.

→ Dato $c_A, d_A, n_A, e_A, n_A = p \cdot q$

- si precalcolano i valori $d_A \bmod (p-1)$

  $d_A \bmod (q-1)$

  $k = p^{-1} \bmod q$

- si imposta ~~il~~ l'algoritmo come:

1. $v_1 = c_A^{[d_A \bmod (p-1)]} \bmod p$
2. $v_2 = c_A^{[d_A \bmod (q-1)]} \bmod q$
3. $u = (v_2 - v_1)\, k \bmod q$
4. $M_{chiaro} = (v_1 + pu) \bmod (p \cdot q)$.

Verifica di validità:

~~(mod q)~~ (mod q)

$v_1 + pu \overset{\checkmark}{=} v_1 + p(v_2 - v_1)\, k \bmod q = v_2 \bmod q$.

$v_1 + pu \bmod p = v_1 \bmod p$

quindi la soluzione M_{chiaro} è effettivamente il Testo in chiaro cercato.

Note di Crittografia su curve ellittiche

1 Curve ellittiche

Definizione 1.1. Una curva ellittica[1] E su un campo K è definita come l'insieme dei punti soddisfacenti un'equazione algebrica del tipo:

$$E(K) = \{(x, y) \in K^2 : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,\ a_i \in K\}$$

La curva deve essere priva di punti singolari, cioè ogni punto della curva deve essere dotato di un'unica tangente. La definizione data è valida per ogni campo, ma per gli usi in crittografia si è interessati unicamente ai campi finiti e in particolare soltanto ai campi finiti $K = F_q \quad q = p^n$ con caratteristica $char(F_q) = 2$ o $char(F_q) = p$ con p un numero primo *grande*. Se $char(F_q) = p > 3$ allora attraverso il seguente cambio di variabili:

$$X \leftarrow X - \frac{a_2}{3}$$
$$Y \leftarrow Y - \frac{a_1X + a_3}{2}$$

ponendo $a = \frac{1}{9}a_1^2 + a_4$, $b = \frac{2}{27}a_2^3 - \frac{1}{3}a_2a_4a_6$ si dimostra l'isomorfismo con la curva:

$$y^2 = x^3 + ax + b \quad x, y \in F_p$$

perché non ci siano punti singolari, in tutti i punti deve esistere un'unica tangente alla curva; quindi se $y^2 = f(x)$, $f'(x) = 2y\frac{dy}{dx}$. L'espressione $\frac{dy}{dx}$ risulta indefinita se e solo se $f'(x_0) = f(x_0) = y_0 = 0$. Detto in altri termini l'equazione di terzo grado $f(x) = x^3 + ax + b$ non deve avere radici multiple e questo succede soltanto se il suo discriminante $D = 4a^3 + 27b^2 \neq 0$.
Nel caso in cui $char(F_q) = 2$, assumendo $a_1 \neq 0$ con il seguente cambio di variabili

$$X \leftarrow a_1^2X - \frac{a_3}{a_1}$$
$$Y \leftarrow a_1^3Y - \frac{a_1^2a_4 + a_3^2}{a_1^3}$$

si ottiene un'equazione della forma:

$$y^2 + xy = x^3 + ax^2 + b, \quad a, b \in F_{2^n}; \quad b \neq 0$$

e con considerazioni analoghe alle precedenti si trova che la condizione di non-singolarità corrisponde a garantire $b \neq 0$. Per definire un crittosistema utilizzando i

[1] Tutti i simboli = sono da intendersi come $\equiv$; mentre i simboli di derivata sono intesi come derivate formali su un polinomio.

punti di una curva ellittica, è necessario dotare tale insieme di una struttura algebrica. La più semplice struttura che fornisce tutti gli strumenti necessari è la struttura di gruppo. Definiamo quindi una legge di composizione interna tra due punti di una curva ellittica che sia associativa, ammetta un elemento neutro e tale che sia definito l'inverso di ogni punto. Si può dimostrare che un'operazione definita come mostrato di seguito attribuisce all'insieme dei punti della curva la struttura di un gruppo abeliano $(E, +)$.

Immaginiamo l'elemento neutro $\{O\}$ della nostra operazione come un punto del piano infinitamente distante dall'origine, situato lungo l'asse delle ordinate; vedremo in seguito come $\{O\}$ si identifichi con il punto della curva coincidente con il punto improprio del piano proiettivo corrispondente alla direzione verticale.

Definizione 1.2 (Legge di composizione)**.** Siano $P, Q \in E(K)$ due punti della curva, $r = \overline{PQ}$ la retta passante per P e Q oppure tangente alla curva se $P = Q$ e R l'ulteriore punto d'intersezione di r con $E(K)$. Detta $r' = \overline{RO}$ la retta verticale passante per R, si definisce $P + Q$ uguale all'ulteriore punto di intersezione tra r' e $E(K)$.

La legge di composizione enunciata è equivalente alla proposizione:
$P, Q, R \in E(K) \ : \ P + Q + R = O \iff P, Q, R$ sono allineati.

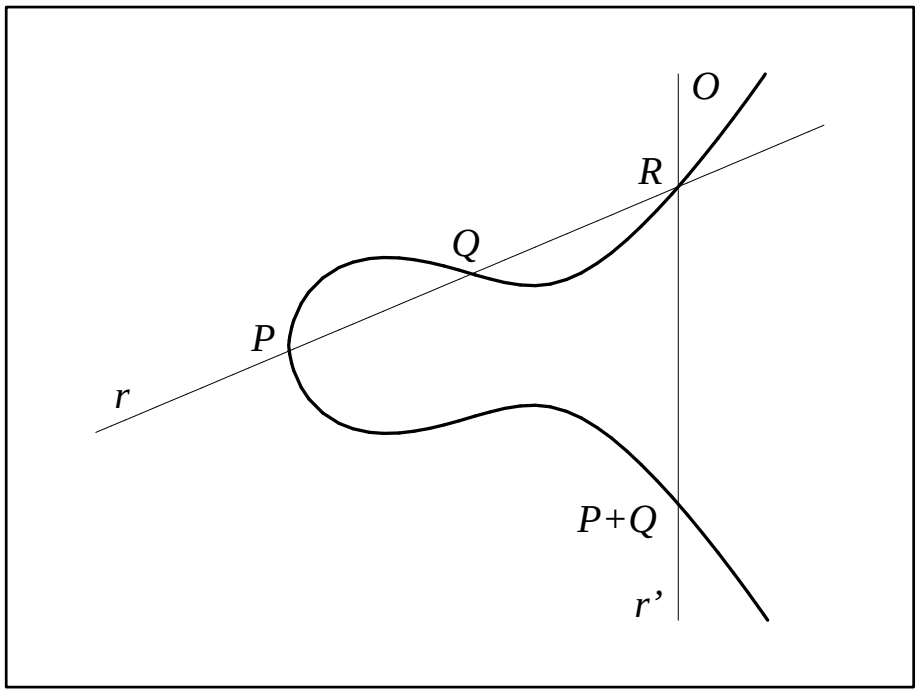


Figura 1: Somma di punti, $K = \mathbb{R}$ $y^2 = x^3 - 3x + 6$.

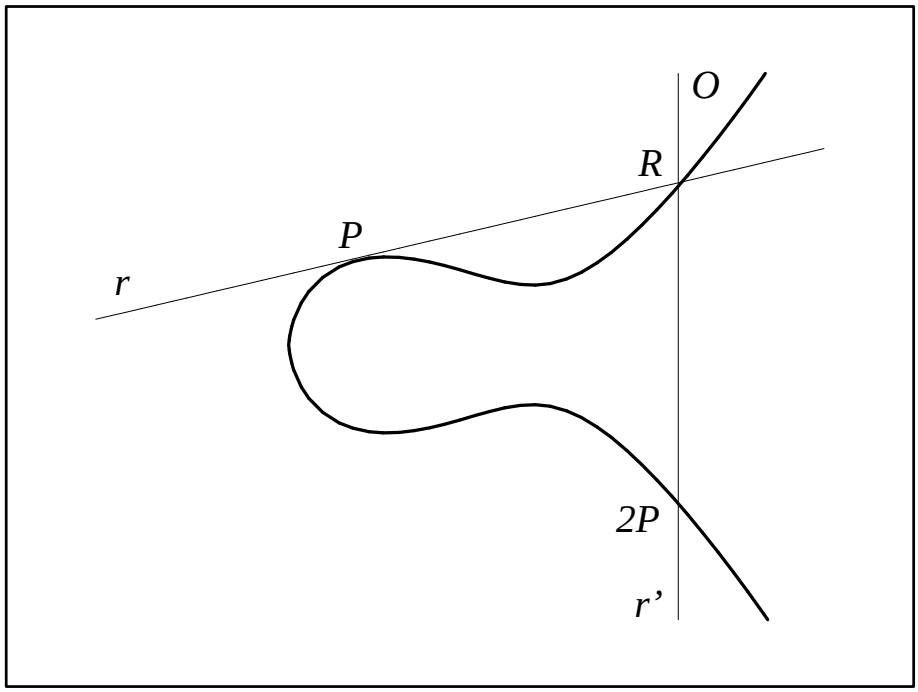


Figura 2: Raddoppio di un punto, $K = \mathbb{R}$ $y^2 = x^3 - 3x + 6$.

La legge di composizione enunciata attribuisce all'insieme dei punti della curva la struttura di gruppo abeliano additivo, dove l'elemento neutro è rappresentato dal punto O, valgono, infatti, e seguenti proprietà:

- $\forall P \in E(K),\ P + O = P$;
- $\forall P \in E(K),\ \exists\,!\, Q = (-P) \in E(K) : P + Q = O$;
- $P + (Q + R) = (P + Q) + R$, con $P, Q, R \in E(K)$;
- $P + Q = Q + P$, con $P, Q \in E(K)$;

Nel caso in cui il campo K coincida con l'insieme dei numeri reali $\mathbb{R}$ è possibile fornire una rappresentazione grafica delle operazioni definite sulle curve, come mostrato nelle fig. 1 e 2. Nel caso dei campi finiti l'insieme dei punti della curva conserva le stesse proprietà algebriche pur non essendo più possibile una loro rappresentazione grafica.

In formule, dati $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, $P_3 = P_1 + P_2 = (x_3, y_3)$, $P_4 = 2P_1 = (x_4, y_4)$ le espressioni esplicite della legge di composizione sono indicate in tab.1 e 2

$P_1 + P_2$	$x_3 = (\frac{y_1-y_2}{x_1-x_2})^2 - x_1 - x_2$ $y_3 = (\frac{y_1-y_2}{x_1-x_2})(x_1 - x_3) - y_1$
$2P_1$	$x_4 = (\frac{3x_1^2+a}{2y_1})^2 - 2x_1$ $y_4 = -y_1 + (\frac{3x_1^2+a}{2y_1})(x_1 - x_4)$
$-P_1$	$(x_1, -y_1)$

Tabella 1: Formule di addizione in coordinate affini, $char(K) = p > 3$.

Dim. λ costituisce il coefficiente angolare della retta r congiungente P_1 e P_2 che avrà equazione $y = \lambda(x - x_1) + y_1$. Secondo la definizione di somma occorre trovare il punto di intersezione di r con la curva E. Sostituendo nell'equazione della curva, otteniamo un polinomio monico di terzo grado: $x^3 - \lambda^2 x + ... = 0$ di cui conosciamo già due soluzioni(x_1, x_2; le ascisse dei punti P_1 e P_2). Ricordando che la somma delle tre radici di un polinomio monico di grado n è uguale all'opposto del coefficiente di grado $n - 1$ si ha che $x_3 = \lambda^2 - x_1 - x_2$; per trovare poi y_3 basta sostituire x_3 nell'equazione della retta e prendere l'opposto del risultato. Per quanto riguarda la formula del raddoppio, l'unica differenza è nel calcolo del coefficiente angolare della retta che può essere ricavato derivando formalmente l'equazione della curva. Per determinare l'inverso di un punto P basta intersecare la curva con la retta verticale passante per P.

□

$P_1 + P_2$	$x_3 = (\frac{y_1+y_2}{x_1+x_2})^2 + (\frac{y_1+y_2}{x_1+x_2}) + x_1 + x_2 + a$ $y_3 = (\frac{y_1+y_2}{x_1+x_2})(x_1 + x_3) + x_3 + y_1$
$2P_1$	$x_4 = x_1^2 + \frac{b}{x_1^2}$ $y_4 = x_1^2 + (x_1 + \frac{y_1}{x_1})x_3 + x_3$
$-P_1$	$(x_1, x_1 + y_1)$

Tabella 2: Formule di addizione in coordinate affini, char(K) = 2.

Dim. La dimostrazione è del tutto analoga a quella del caso precedente, con l'unica avvertenza che i coefficienti dei polinomi vanno ora intesi modulo 2: $x_3 = (\lambda^2 + \lambda + a) + x_1 + x_2$.

□

2 Coordinate proiettive

Per piano proiettivo su un campo K si intende l'insieme delle classi di equivalenza delle t-uple (X,Y,Z), con non tutte le componenti nulle; per due t-uple equivalenti si ha:

$$(X', Y', Z') \sim (X, Y, Z) \text{ se } (X', Y', Z') = (\lambda X, \lambda Y, \lambda Z),\ \lambda \in K$$

Ogni classe di equivalenza è chiamata punto proiettivo. Se un punto proiettivo ha Z non nullo, allora esiste un'unica t-upla nella sua classe di equivalenza uguale a $(x, y, 1)$, impostando $x = X/Z$ e $y = Y/Z$. Il piano proiettivo può dunque essere identificato con il piano affine (x, y) con l'aggiunta dei punti per i quali $Z = 0$. Questi ultimi sono chiamati punti all'infinito e informalmente possono essere visualizzati come l'orizzonte del piano affine. Si osserva come il punto all'infinito di una retta propria nel piano proiettivo sia $(1, m, 0)$, quindi indicativo della direzione di tale retta. Ad ogni equazione $F(x, y) = 0$ di una curva nel piano affine corrisponde un'equazione $F(X, Y, Z) = 0$. Se si applica la precedente sostituzione all'equazione di una curva ellittica si ottiene un'equazione omogenea di terzo grado:

$$Y^2Z = X^3 + aXZ^2 + bZ^3, \text{ se } char(K) > 3$$

$$Y^2Z + XYZ = X^3 + aX^2Z + bZ^3, \text{ se } char(K) = 2$$

Cercando i punti di intersezione delle precedenti curve con la retta all'infinito $Z = 0$ si individua il punto proiettivo $(0, 1, 0)$ che corrisponde alla direzione delle rette verticali. Quest'ultimo è il punto assunto come elemento neutro nella legge di composizione enunciata nel paragrafo precedente.

Nelle implementazioni pratiche dei crittosistemi, utilizzare coordinate proiettive comporta, nella marggior parte dei casi, vantaggi computazionali significativi. Infatti, utilizzando un tale sistema di coordinate, le formule di addizione, dei punti sulla curva, non presentano alcuna operazione di inversione, che costituisce l'operazione aritmetica computazionalmente più onerosa. A titolo di esempio si riportano le formule di addizione nel caso di una curva con $char(K) \neq 2, 3$.

$$P_1 = (X_1, Y_1, Z_1),\ P_2 = (X_2, Y_2, Z_2),\ P_3 = P_1 + P_2 = (X_3, Y_3, Z_3)$$

$$\begin{cases} X_3 = p_1(t - q_2) \\ 2Y_3 = r(3q_2 - 2t) - p_3(s_2 + s_1) & \text{se } (P_1 \neq P_2) \\ Z_3 = wp_3 \end{cases}$$

$$
\begin{array}{ll}
u_1 = X_1 Z_2 & p_2 = p_1^2 \\
u_2 = X_2 Z_1 & p_3 = p_1 p_2 \\
s_1 = Y_1 Z_2 & q_1 = u_1 + u_2 \\
s_2 = Y_2 Z_1 & q_2 = p_2 q_1 \\
w = Z_1 Z_2 & r = s_2 - s_1 \\
p_1 = u_1 - u_2 & t = w r^2
\end{array}
$$

$$
\begin{cases}
X_3 = \lambda_1(\lambda_4 - 4\lambda_5) \\
Y_3 = \lambda_2(6\lambda_5 - \lambda_4) - 2Y_1^2\lambda_6 & \text{se } (P_1 = P_2) \\
Z_3 = \lambda_1\lambda_6
\end{cases}
$$

$$
\begin{array}{ll}
\lambda_1 = 2Z_1Y_1 & \lambda_4 = \lambda_2^2 \\
\lambda_2 = 3X_1^2 + aZ_1^2 & \lambda_5 = \lambda_1\lambda_3 \\
\lambda_3 = X_1Y_1 & \lambda_6 = \lambda_1^2
\end{array}
$$

3 Ordine del gruppo di punti di una curva ellittica

Ipotizziamo, da ora in poi, di lavorare con un campo finito $GF(p)$, con p un numero primo *grande*. Daremo adesso un semplice metodo di conteggio, che permette di stabilire un range di variabilità per l'ordine del gruppo di punti di una curva $E(GF(p))$. Occorre individuare un metodo per cui dato $x \in GF(p)$, si possa stabilire se y è un residuo quadratico modulo p, cioè se $y^2 = x^3 + ax + b \, mod \, p$.

Definizione 3.1. Sia a un intero e $p > 2$ un numero primo. Il simbolo di Legendre è definito come segue.

$$
\left(\frac{a}{p}\right) = \begin{cases}
0, & \text{se} \quad p \mid a \\
1, & \text{se} \quad \exists \alpha : \alpha^2 \equiv a \, (mod \, p) \\
-1, & \text{se} \quad \nexists \alpha : \alpha^2 \equiv a \, (mod \, p)
\end{cases}
$$

Per poter calcolare il simbolo di Legendre si usa la seguente:

Proposizione 3.1.

$$
\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \, (mod \, p)
$$

Dim. Il caso quando $p \mid a$ è immediato. Sia dunque $a > 0$ e $p \nmid a$. Per il *piccolo teorema di Fermat* si ha che $(a^{(p-1)/2})^2 \equiv 1 \, (mod \, p)$ quindi $a^{(p-1)/2} \equiv \pm 1 \, (mod \, p)$. Sia $g \in GF(p)^*$ il generatore del gruppo moltiplicativo, allora $a = g^j$ è un residuo

quadratico se e solo se j è pari. $a^{(p-1)/2} \equiv g^{j(p-1)/2} \equiv 1 \,(mod\, p)$ se e soltanto se $(j(p-1)/2)$ è divisibile per $(p-1)$ e ciò avviene se e soltanto se j è pari. In conclusione, quindi, $a^{(p-1)/2} \equiv \pm 1 \,(mod\, p)$ ed è uguale a 1 se e solo se a è un residuo quadratico.

□

Tornando alla curva di equazione $y^2 = f(x) = x^3 + ax + b$ con $a, b \in GF(p)$, si ha:

- il punto all'infinito $O = (0 : 1 : 0)$ è un punto della curva.
- se $f(x)$ è un residuo quadratico allora ci sono due punti: $(x, \pm y)$
- se $p \mid f(x)$ allora c'è il solo punto $(x, 0)$
- se $f(x)$ non è un residuo quadratico, non ci sono punti con ascissa uguale a x.

Riassumendo si ha:

$$\mid E\,(GF(p))\mid = 1 + \sum_{x \in GF(p)} \left(1 + \left(\frac{f(x)}{p}\right)\right) = p + 1 + \sum_{x \in GF(p)} \left(\frac{f(x)}{p}\right).$$

i limiti inferiore e superiore che è possibile quindi individuare per l'ordine della curva sono:

$$1 \leq \mid E(GF(p))\mid \leq 2p + 1$$

Esempio 3.1. *sia* $E : y^2 = x^3 + x + 6$ *su* $GF(11)$*, allora* $p = 11 \equiv 3 \,mod\, 4$*, per determinare l'ordine della curva applichiamo la formula:*

$$\mid E \mid = p + 1 + \sum_{x \in GF(11)} \left(\frac{x^3 + x + 6}{11}\right).$$

x	$f(x) = x^3 + x + 6 \,mod\, 11$	$(f(x)/p)$	(x, y)
0	6	-1	
1	8	-1	
2	5	$+1$	$(2,4); (2,7)$
3	3	$+1$	$(3,5); (3,6)$
4	8	-1	
5	4	$+1$	$(5,2); (5,9)$
6	8	-1	
7	4	$+1$	$(7,2); (7,9)$
8	9	$+1$	$(8,3); (8,8)$
9	7	-1	
10	4	$+1$	$(10,2); (10,9)$

L'ordine della curva vale $\mid E \mid = p + 1 + 1 = 13$*, quindi il gruppo dei punti della curva è ciclico e in particolare ogni suo elemento è generatore.*

Per campi con p *grande*, nelle implementazioni pratiche, il calcolo esatto del numero di punti fà ricorso ad algoritmi specifici a seconda della caratteristica del campo, oppure al metodo di Schoof.

In generale per curve definite su un campo finito arbitrario $GF(q)$ $q = p^m$, valgono i seguenti risultati:

Teorema 3.1 (di Hasse)**.** *Sia E una curva ellittica su $GF(q)$. Allora*

$$\big|\#E(GF(q)) - (q+1)\big| \leqslant 2\sqrt{q}.$$

Si vede, quindi, come la cardinalità del gruppo $E\big(GF(q)\big)$ sia circa uguale alla cardinalità del campo $GF(q)$ su cui la curva è definita. A differenza del gruppo moltiplicativo $GF(q)^*$, il gruppo additivo $E\big(GF(q)\big)$ non è necessariamente ciclico, cioè non ammette necessariamente un unico generatore. La sua struttura è precisata dal seguente:

Teorema 3.2. *$E\big(GF(q)\big)$ è un gruppo abeliano di rango[2] tipo (n_1, n_2), ossia*

$$E\big(GF(q)\big) \cong \mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2}$$

dove n_2 divide n_1. Inoltre n_2 divide $q-1$.[3]

Corollario 3.1. *Se $\#E\big(GF(q)\big)$ è fattorizzabile nel prodotto di primi distinti, allora il gruppo $E\big(GF(q)\big)$ è ciclico.*

Dim. Se $\#\ E\big(GF(q)\big) = p_1 p_2$, allora $p_1 p_2 = n_1 n_2$, ma n_2 deve dividere n_1, quindi $n_2 = 1$ e poiché $\mathbb{Z}_{n_1}$ è ciclico deve esserlo necessariamente anche $E\big(GF(q)\big)$ □

Per poter immergere il problema del logaritmo discreto nel gruppo di punti di una curva ellittica abbiamo ancora bisogno di definire un metodo per trovare un punto di una curva ellittica e precisamente un generatore di un sottogruppo ciclico del gruppo dei punti della curva. Si considereranno quindi curve con ordine uguale ad un numero primo, oppure che ammettano un fattore primo molto grande: $\mid E(GF(p)) \mid = c \cdot p_1$ con $p_1 \approx 2^{160}$, in questo caso per ricondursi a lavorare nel gruppo ciclico di ordine p_1, cioè per trovare un suo generatore, si estrae un punto a caso $P(x, y)$ e si moltiplica scalarmente tale punto per il co-fattore c, incrementando il valore di x e iterando il procedimento finché non si ottiene un punto diverso dall'elemento neutro.

[2]Dato un gruppo finito K si definisce rank(K) il numero minimo di elementi generatori individuabili nell'insieme.

[3]Si sottolinea come $\mathbb{Z}_{n_1}$ e $\mathbb{Z}_{n_2}$ indichino i gruppi ciclici additivi dei residui modulo n_1 e n_2, rispettivamente.

Supponendo di lavorare con curve definite su campi $GF(p)$ $p > 3$, affrontiamo adesso il problema di trovare un punto su una curva di equazione assegnata.

Il metodo più semplice che è possibile immaginare consiste nel prendere casualmente un valore per la coordinata x e tentare di estrarre la radice quadrata di $y^2 = f(x)$ modulo p affinché il punto di coordinate (x, y) soddisfi l'equazione della curva. Dobbiamo quindi individuare un metodo per estrarre la radice quadrata di un numero $y^2 \equiv z \,(mod\, p)$.

Se $p \equiv 3 \,mod\, 4$ allora $y = z^{(p+1)/4} \,(mod\, p)$ infatti se z è davvero un residuo quadratico, applicando il piccolo di Fermat si ha: $y^2 = z^{(p+1)/2} = z\, z^{(p-1)/2} = z \,(mod\, p)$.

Se $p \equiv 1 \,mod\, 4$ allora il metodo sicuramente più utilizzato è l'algoritmo di Tonelli-Shanks per i cui dettagli si rimanda alla bibliografia. Di seguito è riportata una versione alternativa dell'algoritmo (non ottimizzata).
Essendo p un numero dispari si può scrivere

$$\mathbf{p - 1 = 2^h t} \text{ con } t \text{ dispari}, h \geq 2$$

Si sceglie un elemento u che non sia un residuo quadratico, cioè: $u^{(p-1)/2} \equiv -1(mod p)$, quindi si pone $\mathbf{v = u^t}$, si può dimostrare che v è un radice primitiva 2^h-esima dell'unità in $GF(p)$, cioè genera l'unico sottogruppo G con ordine 2^h in $GF(p)^*$.

Dim. Dire che $v = u^t$ è una radice 2^h-esima dell'unità equivale a provare che $t2^h \equiv 0 \,mod\, (p-1)$, che è sicuramente vero viste le ipotesi. Per dimostrare che è anche un elemento primitivo procediamo per assurdo supponendo che non lo sia. Quindi v avrà un ordine m uguale ad un divisore di 2^h, cioè $m \mid 2^h$ e $v^m \equiv 1 mod p$. v è esprimibile come una potenza di un elemento primitivo radice 2^h-esima dell'unità cioè $v = g^j$ con $j \mid m$ allora j è pari e v è un residuo quadratico in $GF(p)$. Ricordando che $v = u^t$ di ha:

$$1 = \left(\frac{v}{p}\right) = \left(\frac{u}{p}\right)^t = -1$$

Contraddizione!

□

Dunque il problema è quello di risolvere $\mathbf{y^2 = z \bmod p}$ quindi postuliamo che z sia un residuo quadratico.

- Si osserva che $\mathbf{z^t}$ ha ordine moltiplicativo un divisore di 2^{h-1} perché $(z^t)^{2^{h-1}} = y^{2^h t} = y^{p-1} = 1 \,mod\, p$. Si ha: $z^t \in G$, $G \leq GF(p)^*$, $|G| = 2^h$.
- v è un elemento primitivo del sottogruppo G, quindi ha ordine 2^h
- si può impostare l'equazione: $z^t = v^x$ con $0 \leq x < 2^h$, x pari.

- Estraiamo il logaritmo discreto nel sottogruppo ciclico di ordine 2^h come descritto dall'algoritmo RDL (molto efficiente proprio per la particolarità dell'ordine), ottenendo $\mathbf{x} = \mathbf{log_v^D(z^t)}$.
- impostiamo $\mathbf{k} = \mathbf{v^{x/2}}$ ottenendo $k^2 = z^t$, cioè k è la radice quadrata di z^t.
- Osservando che t dispari si può sempre scrivere come $t = 2m + 1$ la soluzione può così essere costruita come:$\mathbf{y} = \mathbf{kz^{-m}} =$, infatti:

$$\mathbf{y} = (\mathbf{kz^{-m}})^{\mathbf{2}} = k^2 z^{-2m} = v^x z^{-2m} = z^t z^{-2m} = z^{t-2m} = z$$

3.1 Algoritmo per l'estrazione del log discreto in gruppi ciclici di ordine 2^h, $h \geq 1$.

Supponiamo che $v \in GF(p)^*$ sia il generatore di un sottogruppo G tale che $|G| = 2^h$ con $h \geq 1$. Dato un elemento $\alpha \in G$ vogliamo calcolare $x = log_v^D(\alpha)$; Il metodo descritto nel seguito consente di costruire la soluzione a partire dalla soluzione del DL in un gruppo ciclico di ordine 2, dove la questione è banale e il logaritmo è uguale ovviamente a 1. Supponiamo quindi che $h > 1$, si sceglie un intero $0 \leq f < h$ avendo che $\alpha = v^x$ con $0 \leq x < 2^h$, possiamo scrivere $x = q_x 2^f + r_x$ dove $0 \leq r_x < 2^f$ e $0 \leq q_x < 2^{h-f}$ e perciò: $\alpha^{2^{h-f}} = v^{x2^{h-f}} = v^{q_x 2^h + r_x 2^{h-f}} = v^{r_x 2^{h-f}}$. Si osservi come $v^{2^{h-f}}$ abbia ordine 2^f, possiamo quindi ripetere ricorsivamente lo stesso procedimento per risolvere $\alpha^{2^{h-f}} = (v^{2^{h-f}})^{r_x}$ e trovare r_x. Si osservi ancora come $\alpha v^{-r_x} = v^x v^{-r_x} = v^{q_x 2^f} = (v^{2^f})^{q_x}$, di nuovo v^{2^f} ha ordine moltiplicativo 2^{h-f} e quindi ricorsivamente possiamo risolvere l'equazione $\alpha v^{-r_x} = (v^{2^f})^{q_x}$ e trovare q_x. A tal punto rimettendo assieme tutti i risultati trovati si ha $x = q_x 2^f + r_x = log_v^D(\alpha)$.

Algoritmo Log Discreto in gruppi ciclici con ord = 2^h

---

input: f,$v, \alpha \in G$ con $|G| = 2^h$
output: $x = log_v^D(\alpha)$

1. if $h = 1$, return $log_v^D(\alpha)$ /* ricerca esaustiva, gruppo di ordine 2 */
2. else
3. select $f \in 1, ..., h-1$
4. $r_x \leftarrow RDL(p, 2^h, f, v^{2^{h-f}}, \alpha^{2^{e-f}})$ /* $0 \leq r_x < 2^f$ */
5. $q_x \leftarrow RDL(p, 2^h, h-f, v^{2^f}, \alpha / v^{r_x})$ /* $0 \leq q_x < 2^{h-f}$ */
6. return $x \leftarrow 2^f q_x + r_x$

Lo stesso algoritmo si applica a gruppi ciclici di ordine q^h con $q > 2$.

4 Problema del logaritmo discreto su curve ellittiche

Essendo l'insieme di punti $E\big(GF(q)\big)$ dotato di una struttura di gruppo abeliano additivo, è possibile riformulare il problema generalizzato del logaritmo discreto. Dati un gruppo $G \leq E\big(GF(q)\big)$ ciclico (quindi con $n = |G|$, un intero $k \geqslant 2$ e i punti $P, Q \in G$, si definisce l'operazione "kP" nel modo seguente:

$$Q = kP = \underbrace{P + P + \ldots + P}_{\text{k volte}}$$

Dati k e P, risulta relativamente semplice calcolare $Q = kP$ servendosi delle formule di addizione. Il metodo base per il calcolo dell'operazione kP è basato sulla rappresentazione binaria di k. Se $k = \sum_{j=0}^{m-1} k_j 2^j$, dove ogni $k_j \in \{0, 1\}$, allora kP può essere calcolato come:

$$kP = \sum_{j=0}^{m-1} k_j 2^j P = 2(\ldots 2(2k_{m-1}P + k_{m-2}P) + \ldots) + k_0 P.$$

Algoritmo Double & Add

---

input: $k = (k_{m-1} \ldots k_1\, k_0)_2$, $P \in E\big(GF(p)\big)$;
output: kP;

1. $Q \leftarrow O$;
2. For i from $m-1$ down to 0 do:
   2.1. $Q \leftarrow 2Q$;
   2.2. If $(k_i = 1)$ $Q \leftarrow Q + P$;
3. Return(Q).

Il numero medio di operazioni sulla curva, è pari rispettivamente a m raddoppi e $m/2$ somme di punti. Esistono svariati algoritmi per il calcolo del kP più efficienti del double & add che riescono a diminuire ulteriormente il numero medio di somme o raddoppi.

Dati P e Q, risulta però computazionalmente improponibile determinare k, se i parametri della curva E sono scelti opportunamente. Tutti gli algoritmi noti per risolvere il logaritmo discreto, nel caso di applicazione al gruppo di una curva ellittica (con ordine senza fattori primi piccoli) hanno complessità computazionale esponenziale. Tutti i sistemi crittografici basati su curve ellittiche utilizzano gruppi ciclici aventi un numero molto grande di punti; infatti, gli standard del NIST (FIPS 186-2) raccomandano gruppi con un numero di punti (primo) compreso tra 2^{163} e 2^{571}, a seconda della curva e del campo su cui si basa.

Si capisce dunque perché i crittosistemi basati su curve ellittiche possano garantire, con chiavi di lunghezza minore, lo stesso livello di sicurezza assicurato dai cifrari basati sul problema della fattorizzazione dei numeri primi (RSA), o del logaritmo discreto sui gruppi moltiplicativi $GF(q)^*$. Un raffronto comparativo tra vari crittosistemi a pari livello di sicurezza (2^{80} operazioni), che illustra la lunghezza delle chiavi nei diversi casi è mostrato nella tabella seguente:

Symmetric	*ECC*	*RSA/DH/DSA*
80	163	1024
128	283	3072
192	409	7680
256	571	15360

4.1 Protocollo Diffie-Hellmann

Nel consueto scenario in cui due interlocutori, A e B, vogliano comunicare utilizzando un algoritmo a chiave simmetrica (DES, AES, ...); il problema di stabilire una chiave segreta comune, non disponendo di un canale sicuro su cui accordarsi, può essere risolto applicando il protocollo seguente:

- B e A concordano pubblicamente sull'impiego di una curva ellittica $E\big(GF(p)\big)$ avente parametri (a, b, p, n, P_{base})
- Entrambi scelgono una propria chiave segreta come un numero casuale: $k_{s,B} \in [1, n-1]$, $k_{s,A} \in [1, n-1]$;
- B calcola $P_B = k_{s,B} P_{base}$ e lo invia ad A;
- A calcola $P_A = k_{s,A} P_{base}$ e lo invia a B;
- Entrambi, utilizzando il messaggio ricevuto, sono in grado di calcolare la chiave di sessione per l'algoritmo simmetrico concordato come: $P_{AB} = P_{BA} = k_{s,A} k_{s,B} P_{base}$.

Nel caso in cui un avversario, intercetti entrambi i messaggi $P_B = k_{s,B} P_{base}$ e $P_A = k_{s,A} P_{base}$, anche conoscendo tutti i parametri della curva ellittica impiegata, grazie all'intrattabilità del problema del logaritmo discreto, non può determinare la chiave di sessione P_{AB}.

4.2 Crittosistema di ElGamal

Supponiamo ora che gli interlocutori, A e B, vogliano comunicare utilizzando come metodo di cifratura il protocollo a chiave pubblica di ElGamal. Fissata una curva ellittica $E\big(GF(p)\big)$ con parametri (a, b, p, n, P_{base}), A e B devono scegliere la propria chiave privata e depositare, in un elenco pubblico, la propria chiave pubblica. Per creare la coppia di chiavi entrambi devono:

- Generare un numero casuale $k_s \in [1, n-1]$, prendendolo come chiave privata.
- Calcolare la chiave pubblica $k_p = k_s P$, rendendola disponibile a tutti.

Se A vuole inviare un messaggio m a B:

- Preleva una copia autentica della chiave pubblica di B $k_{p,B}$, tramite eventualmente un'autorità di certificazione (CA);
- Traduce il messaggio m come un elemento $M \in E(GF(p))$;
- Estrae un numero casuale $r \in [1, n-1]$;
- Invia a B il messaggio cifrato costituito dalla coppia di punti: $(rP_{base},\ M + rk_{p,B})$;

B per decifrare il messaggio M deve:

- Calcolare $rP_{base} \cdot k_{s,B} = rk_{p,B}$;
- Recuperare il messaggio in chiaro calcolando $(M + rk_{p,B}) - (rk_{p,B})$.

4.3 Algoritmo di firma ECDSA (Elliptic Curve Digital Signature Algorithm)

Supponiamo che l'utente A voglia inviare un messaggio firmato a B. Fissati pubblicamente il campo finito $GF(q)$, l'equazione della curva, il suo ordine n e un suo elemento generatore P. La coppia di chiavi in possesso di ogni utente sarà: $k_{priv} = (s), s \in Z_n$ e $k_{pub} = (s \cdot P)$. Per firmare un messaggio $m \in \{0,1\}^*$ (stringa binaria)
A esegue i seguenti passi:

1. Genera un numero casuale $r \in [1, n-1]$ coprimo con n.
2. Calcola $r \cdot P = (x_1, y_1)$, $k = x_1 \, mod \, n$. Se $k = 0$ allora si torna al passo 1.
3. Calcola $r^{-1} \, mod \, n$.
4. Calcola $e = SHA - 1(m)$. (SHA-1(.) è una funzione di hash standard)
5. Calcola $z = r^{-1}(e + sk) \, mod \, n$. Se $z = 0$ allora si torna al passo 1.
6. La firma sul messaggio m è (k, z).

Per verificare firma sul messaggi, B esegue invece i seguenti passi.

1. Verifica che (k, z) siano contenuti nell'intervallo [1,n-1].
2. Calcola $e = SHA - 1(m)$;
3. Calcola $w = z^{-1} \, mod \, n$.
4. Calcola $u_1 = ew \, mod \, n$ e $u_2 = kw \, mod \, n$.
5. Calcola $X = u_1 P + u_2 k_{pub,A} = (x_1, y_1)$. Se $X = O$ allora rifiuta la firma, altrimenti accetta la firma se e soltanto se $x_1 \, mod \, n = k$

Se la firma (k, z) fosse autentica, allora $z = r^{-1}(e + sk) \, mod \, n$, si ha:

$$u_1 P + u_2 k_{pub,A} = ez^{-1} P + kz^{-1}(sP) = (z^{-1}(e + sk))P = rP = (x_1, y_1)$$

e quindi $k = x_1 \, mod \, n$.

Riferimenti bibliografici

[1] Koblitz N., - Algebraic Aspect of Cryptography, Springer, (1999).

[2] Blake I., Seroussi G. and Smart N., - Elliptic Curves in Cryptography. Camebridge University Press, (1999).

[3] Shoup V., - A Computational Introduction to Number Theory and Algebra, beta version: http://www.shoup.net/

Testi sulla crittografia in generale:

[4] Stinson D., - Cryptography Theory and Practice, CRC Press (2002).

[5] Menezes A.J.,Oorschot P.C., Vanstone S.A.,

- Handbook of Applied Cryptography, CRC Press (2001). disponibile su web http://www.cacr.math.uwaterloo.ca/hac/