Una concreta applicazione della crittografia su curve elllittiche

Marco Timpanella

Università degli Studi del Salento

16 Ottobre 2019

smart p@per

Curve ellittiche su $\mathbb{R}$

"It is possible to write endlessly on elliptic curves.
This is not a threat!"
(Serge Lang)

Definizione

Siano $a, b \in \mathbb{R}$ due costanti tali che $4a^3 + 27b^2 \neq 0$.
Una curva ellittica E su $\mathbb{R}$ è il grafo di un'equazione

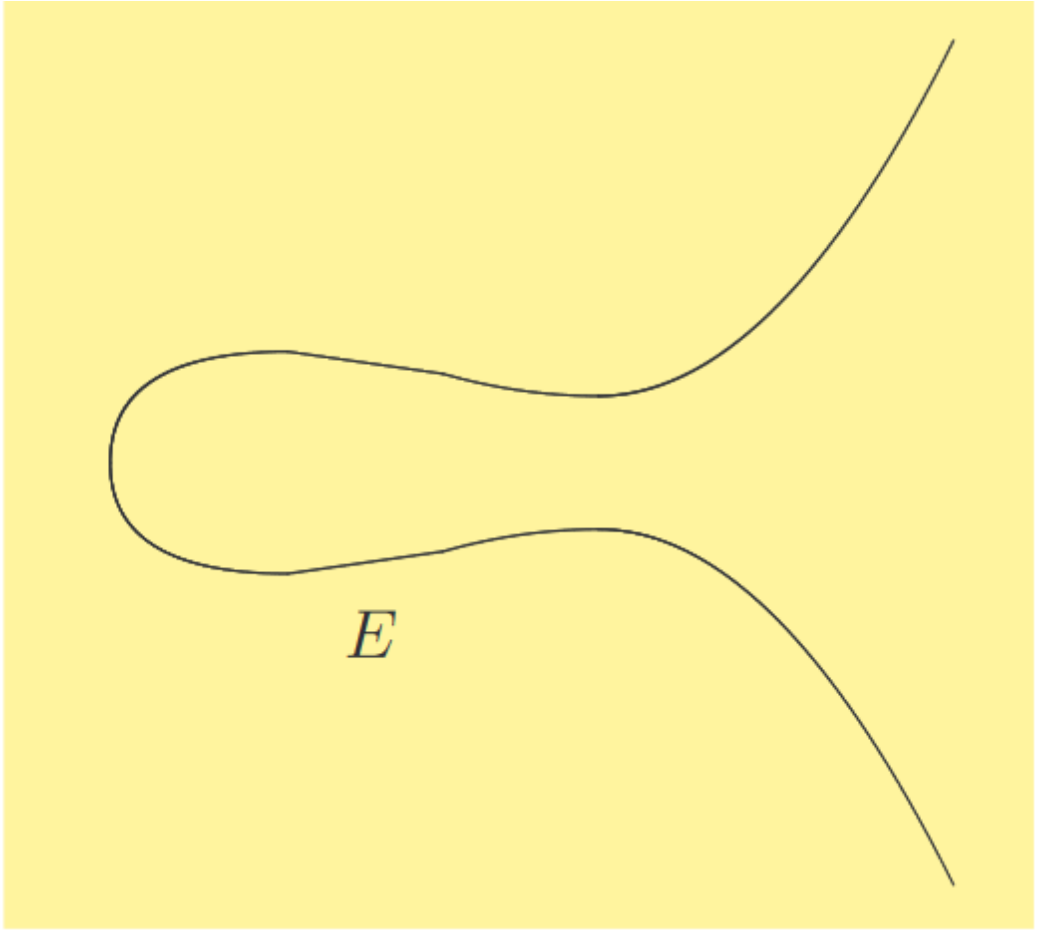
$$Y^2 = X^3 + aX + b,$$

con l'aggiunta di un punto "speciale" ∞, detto punto all'infinito.

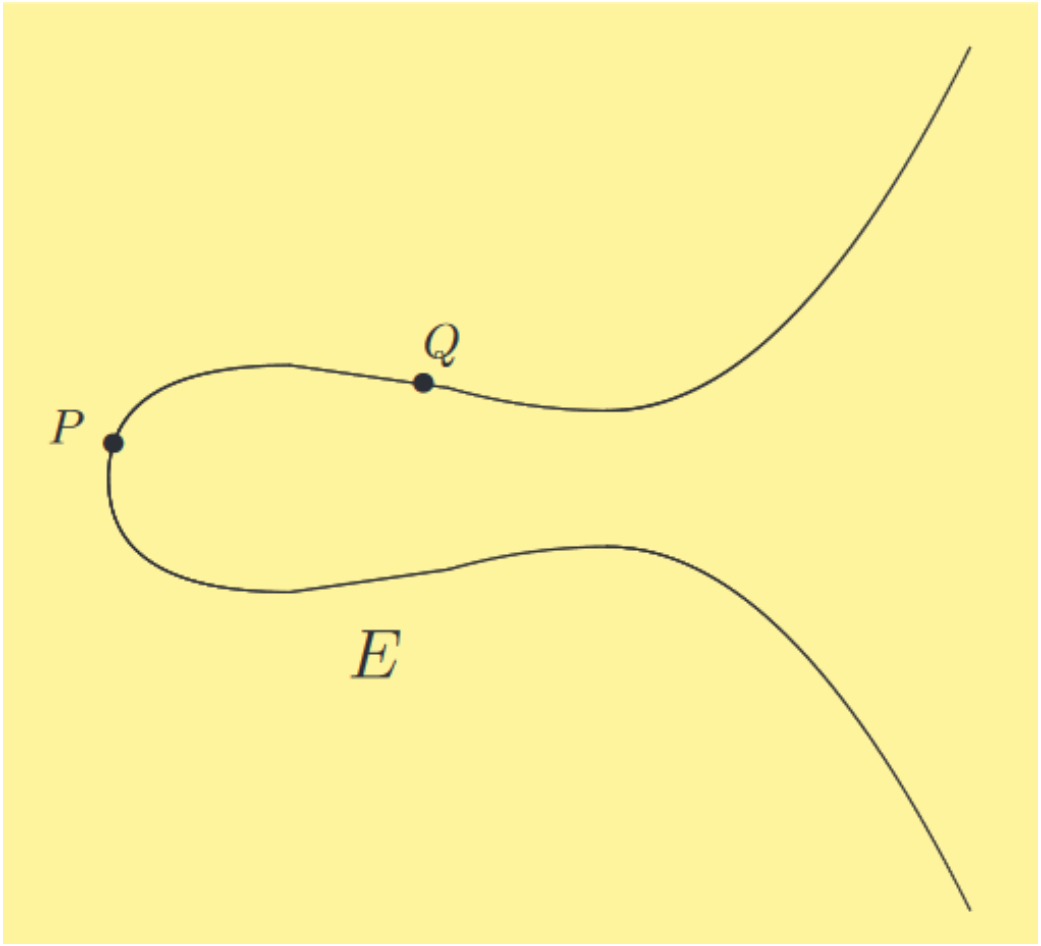
In simboli,

$$E(\mathbb{R}) = \left\{(x, y) \in \mathbb{R}^2 : y^2 = x^3 + ax + b\right\} \cup \{\infty\}.$$

Esempio: $Y^2 = X^3 - 5X + 8$

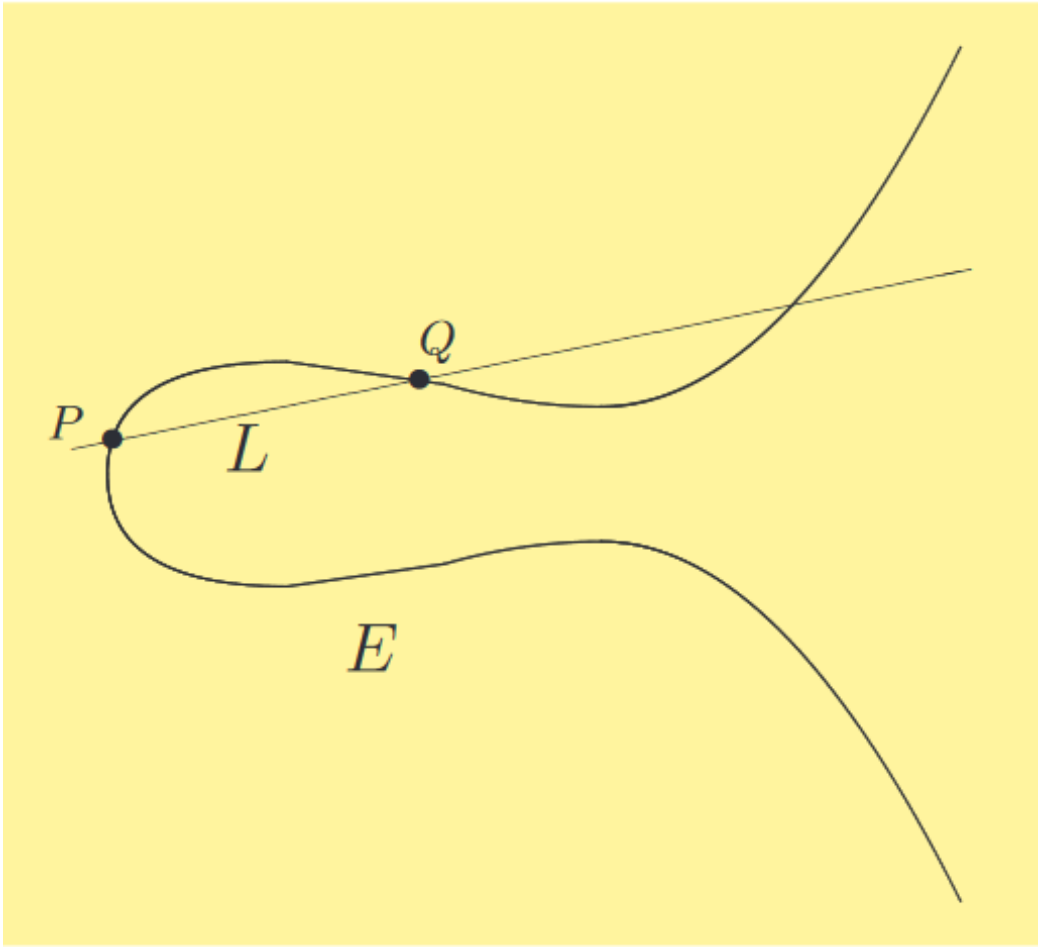


Legge di gruppo



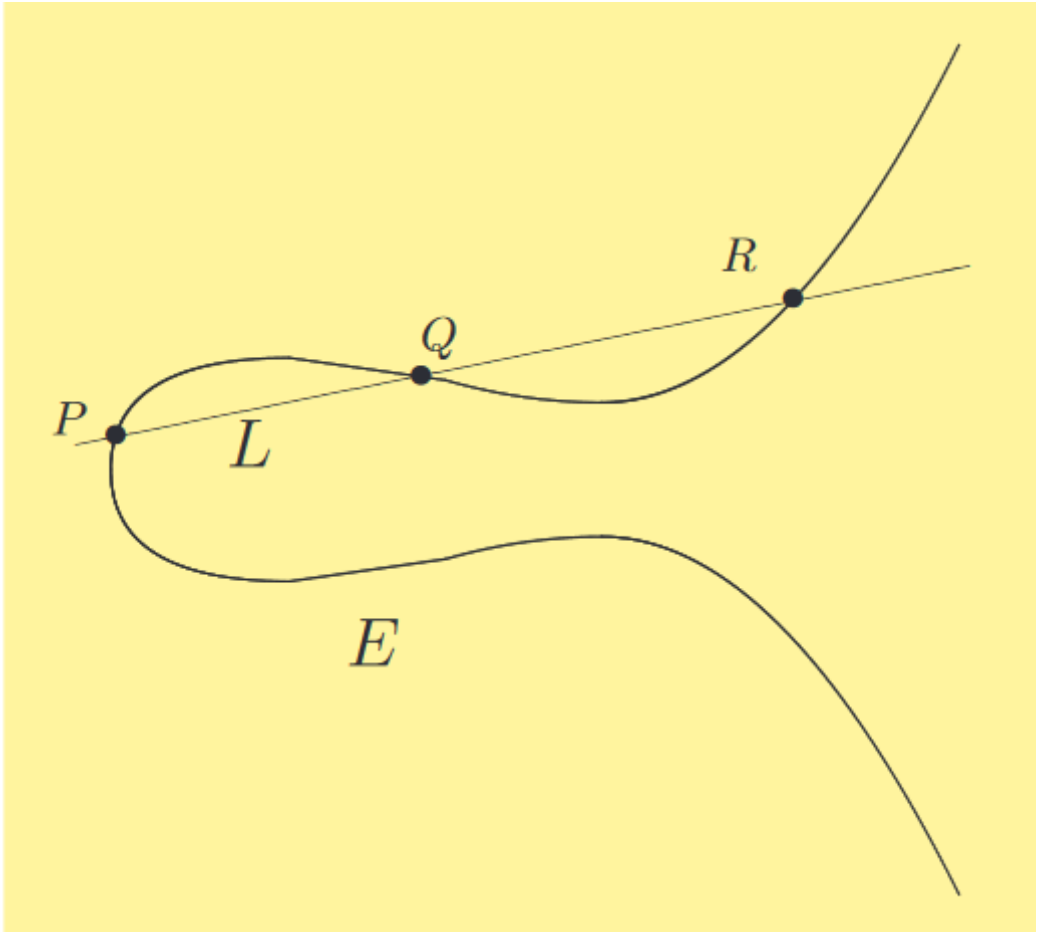
Fissiamo due punti P e Q di E

Legge di gruppo



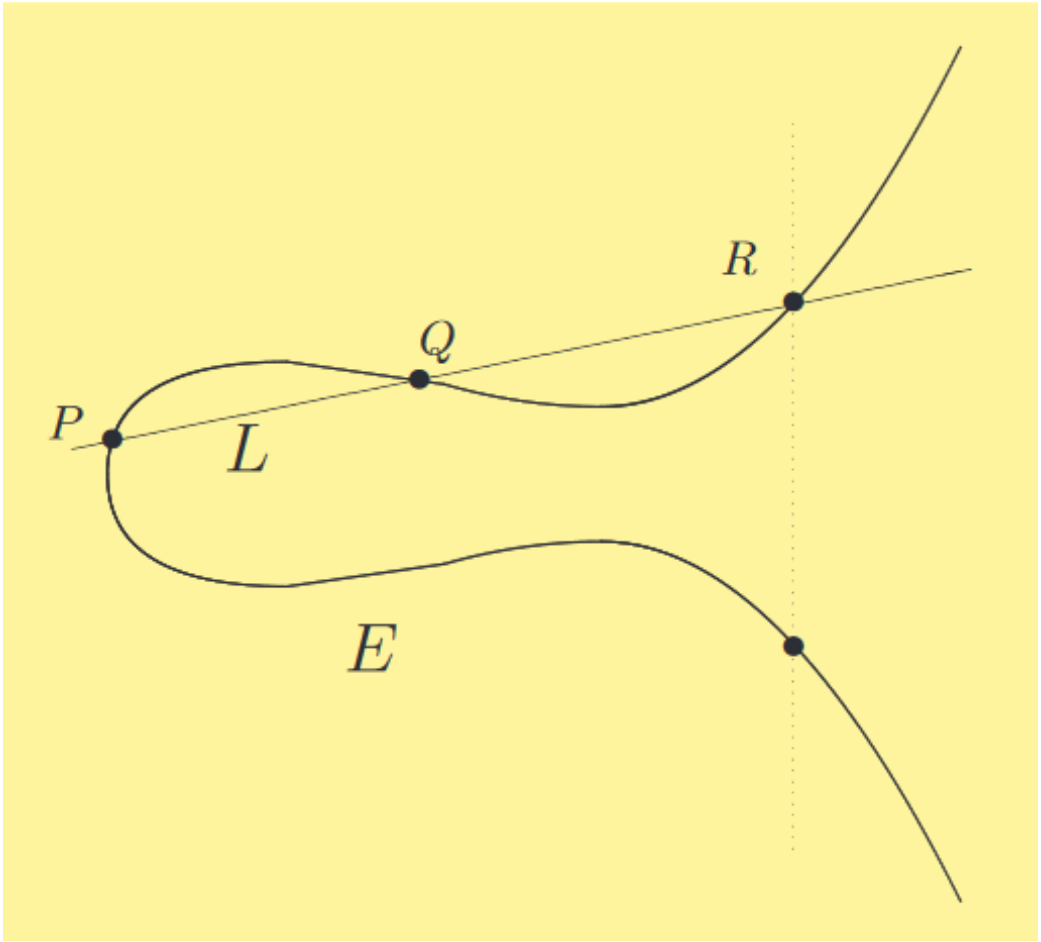
Sia L la retta per P e Q

Legge di gruppo



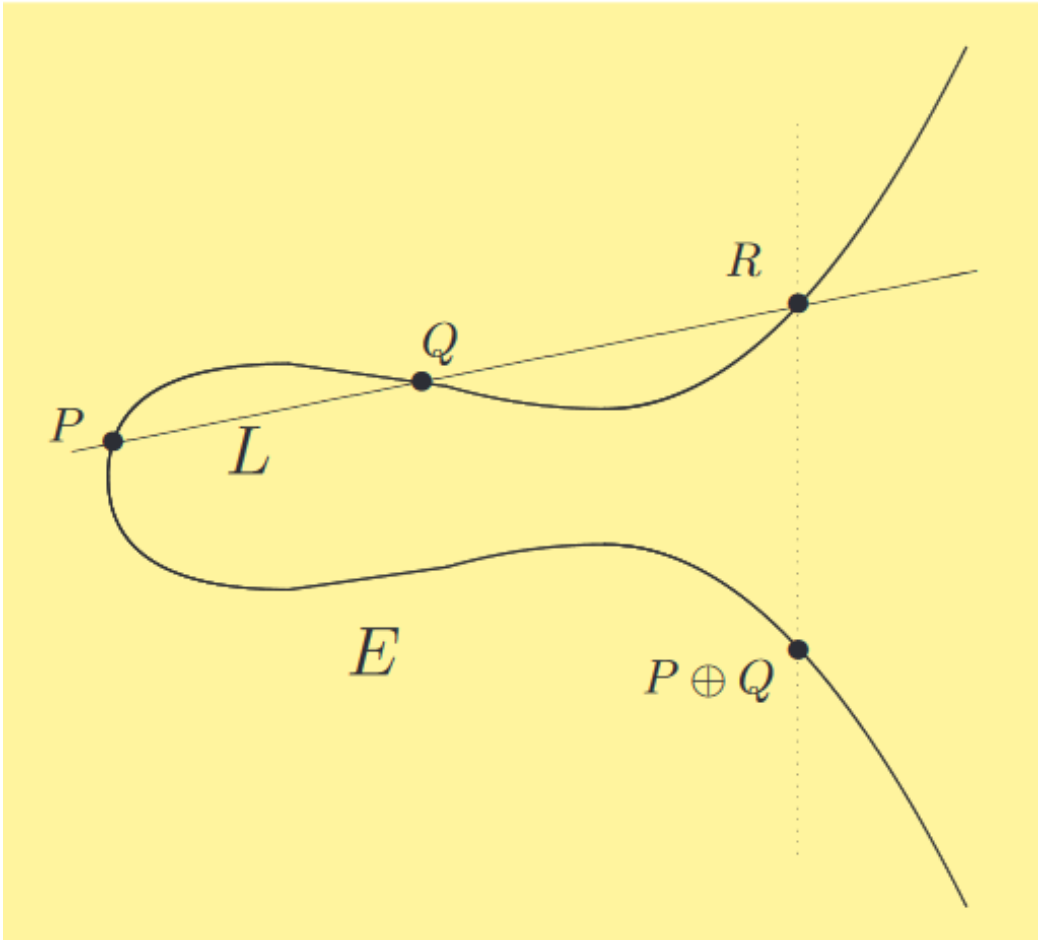
L intersecherà la curva ellittica in un terzo punto R

Legge di gruppo



Consideriamo la retta verticale passante per R

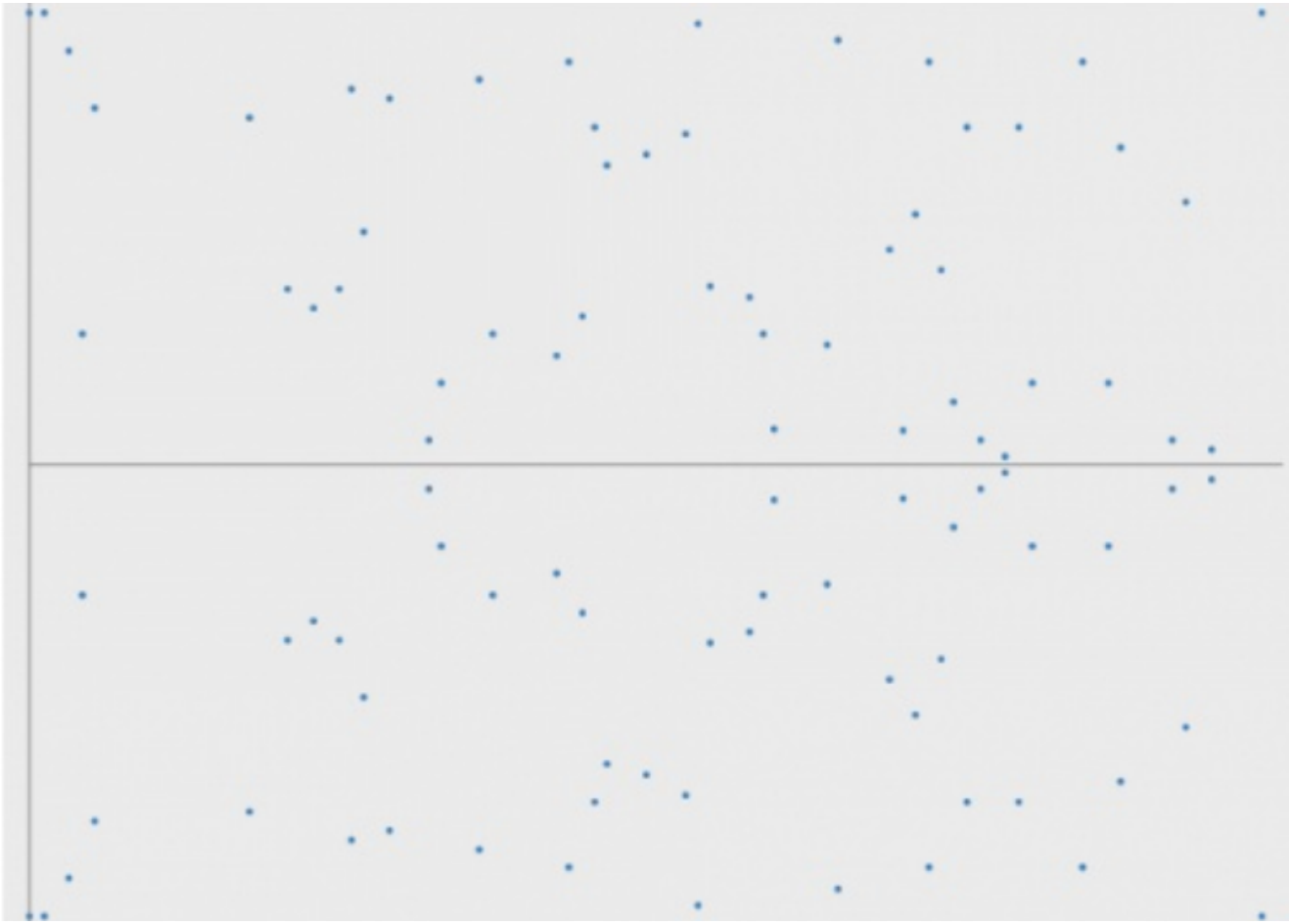
Legge di gruppo



Il punto simmetrico ad R sarà $P \oplus Q$

- Questa operazione è associativa, commutativa e ammette elemento neutro $(\infty) \rightarrow (E(\mathbb{R}), \oplus)$ è un GRUPPO ABELIANO (infinito)
- In crittografia non si usano curve ellittiche su $\mathbb{R}$, ma su campi finiti $\mathbb{F}_q$
- In questo caso, il numero di punti della curva ellittica è finito $(\#E(\mathbb{F}_q) \leq q + 1 + 2\sqrt{q})$
- Curve ellittiche su $\mathbb{F}_q$ possono essere definite "esattamente" come su $\mathbb{R} \rightarrow$ le operazioni su $\mathbb{R}$ vengono sostituite con le analoghe operazioni su $\mathbb{F}_q$.

Stesso esempio, ma su un campo finito



Come prima si può definire $P \oplus Q \rightarrow (E(\mathbb{F}_q), \oplus)$ è un GRUPPO ABELIANO FINITO

Crittografia su curve ellittiche

Utilizzo proposto da Miller e Koblitz nel 1985

Problema del logaritmo discreto (DLP)

$(G, \star)$ gruppo ciclico, $G = \langle g \rangle$. Dato $h \in G$, trovare $a \in \mathbb{N}$ tale che $h = g^a$.

Se ciò di cui abbiamo bisogno è un gruppo, perchè non usare il gruppo di una curva ellittica?

- L'idea generale è di riadattare crittosistemi esistenti alle curve ellittiche
- Da un punto di vista implementativo, le realizzazioni su $\mathbb{F}_{2^n}$ sono più veloci ed economiche
- Su $\mathbb{F}_{2^n}$ una curva ellittica ha equazione

$$Y^2 + XY = X^3 + aX^2 + b$$
$$Y^2 + aY = X^3 + bX + c$$

Pro e contro

Pro

- Stessa sicurezza di RSA o sistemi basati su DLP classico, ma con chiavi più corte (160-256 bit vs 1024-3072 bit)
- L'algoritmo Index Calculus (sub-esponenziale) non è applicabile al gruppo di una curva ellittica

"It is extremely unlikely that an index calculus attack on the elliptic curve method will ever be able to work"
(V.S.Miller)

Contro

- Attacco MOV $\rightarrow$ E non deve essere supersingolare.
  In $\mathbb{F}_{2^n} \longrightarrow Y^2 + XY = X^3 + aX^2 + b$

Problema: Serve un metodo per "trasformare" un messaggio in un punto della curva ellittica.

Idea: Trasformare il messaggio nell'ascissa di un punto della curva ellittica.

Approccio classico: Metodi probabilistici.

- $E : Y^2 + XY = X^3 + aX^2 + b$ curva ellittica su $\mathbb{F}_{2^n}$;
- m messaggio da trasmettere;

Problema: l'equazione $Y^2 + mY = m^3 + am^2 + b$ ammette (almeno) una soluzione?

$Y^2 + \alpha Y + \beta = 0$, $\alpha \neq 0$, ammette soluzione se e solo se $Tr(\beta/\alpha^2) = 0$.

Traccia assoluta

$$Tr : \mathbb{F}_{2^n} \to \mathbb{F}_2 = \{0, 1\}$$

$$x \mapsto \sum_{i=0}^{n-1} x^{2^i} = x + x^2 + x^4 + \ldots x^{2^{n-1}}$$

- $\mathbb{F}_2$-lineare : $Tr(x_1 + x_2) = Tr(x_1) + Tr(x_2)$;

$Y^2 + mY = m^3 + am^2 + b$ ammette (almeno) una soluzione se e solo se

$$Tr(m) + Tr(a) + Tr(b/m^2) = 0.$$

- C'é un 50% di probabilità che $Y^2 + mY = m^3 + am^2 + b$ non ammetta soluzioni ($\rightarrow$ l'algoritmo fallisce!)

Come aumentare le probabilità di successo dell'algoritmo?

- C'é un 50% di probabilità che $Y^2 + mY = m^3 + am^2 + b$ non ammetta soluzioni ($\rightarrow$ l'algoritmo fallisce!)

Come aumentare le probabilità di successo dell'algoritmo?

- $0 \leq m < 2^n/1000$
- $x_j = 1000m + j, \quad 0 \leq j < 1000 \rightarrow m = \lfloor x_j/1000 \rfloor$

Per ogni j, considero l'equazione

$$Y^2 + x_j Y = x_j^3 + a x_j^2 + b.$$

Se ammette soluzione y_j, il messaggio viene mandato nel punto $P = (x_j, y_j)$.

L'algoritmo fallisce se per ogni j l'equazione non ammette soluzione $\rightarrow$ Probabilità 2^{-1000}

Un diverso approccio al problema

$$E_0 : Y^2 + XY = X^3 + a_0X^2 + b_0$$

$$E_1 : Y^2 + XY = X^3 + a_1X^2 + b_1$$

Se per ogni $x \in \mathbb{F}_{2^n} \setminus \{0\}$ vale

$$Tr(a_0) + Tr(b_0/x^2) \neq Tr(a_1) + Tr(b_1/x^2).$$

E_0 ed E_1 si dicono "crittograficamente complementari".

Se E_0 ed E_1 sono crittograficamente complementari, per ogni messaggio $m \in \mathbb{F}_{2^n} \setminus \{0\}$:

$$Tr(m) + Tr(a_0) + Tr(b_0/m^2) \neq Tr(m) + Tr(a_1) + Tr(b_1/m^2).$$

Se E_0 ed E_1 sono crittograficamente complementari, per ogni messaggio $m \in \mathbb{F}_{2^n} \setminus \{0\}$ si presenta sempre uno ed un solo caso tra i seguenti:

(1) $Tr(m) + Tr(a_0) + Tr(b_0/m^2) = 0$

(2) $Tr(m) + Tr(a_1) + Tr(b_1/m^2) = 0$

- Nel caso (1), selezionando la curva ellittica E_0, è possibile trasformare m in un punto $(m, y_m) \in E_0$ ($Y^2 + mY = m^3 + am^2 + b$ ammette una soluzione y_m)
- Viceversa, si seleziona E_1.

Tale selezione viene effettuata per ogni messaggio in transito in una sessione di comunicazione.

Esempio

$$E_0 : Y^2 + XY = X^3 + 1$$

$$E_1 : Y^2 + XY = X^3 + X^2 + 1$$

E_0 ed E_1 sono crittograficamente complementari se e soltanto se per ogni $x \in \mathbb{F}_{2^n} \setminus \{0\}$, vale

$$Tr(1/x) \neq Tr(1) + Tr(1/x),$$

ovvero

$$Tr(1) = 1$$

In $\mathbb{F}_{2^n}$, $Tr(1) = 1$ se e soltanto se n è dispari.

Esempio

$$E_0 : Y^2 + XY = X^3 + 1$$

$$E_1 : Y^2 + XY = X^3 + X^2 + 1$$

- Metodo rapido per il calcolo del numero di punti

$$\#E_j(\mathbb{F}_{2^n}) = 2n + 1 - c_n,$$

$$\begin{cases} c_0 = 2, \\ c_1 = -(-1)^j, \\ c_{r+1} = -(-1)^j c_r - 2c_{r-1} \end{cases}$$

- $\#E_0(\mathbb{F}_{2^n}) = 4p$, $\#E_1(\mathbb{F}_{2^n}) = 2p\prime \rightarrow$ Ottimali contro algoritmo di Pohlig-Helmann

Ulteriori applicazioni

- ECDSA in Bitcoin ed Ethereum;
- SIKE $\rightarrow$ candidato per post-quantum cryptography.

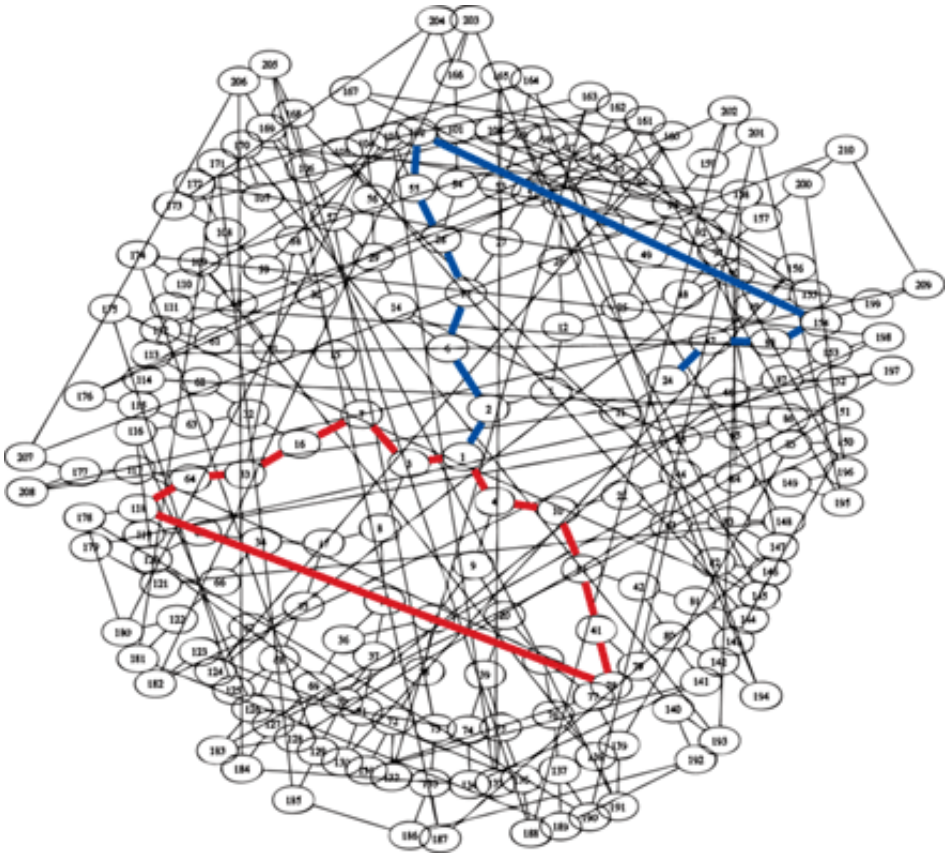


Figure: SIKE (Supersingular Isogeny Key Encapsulation)

GRAZIE PER L'ATTENZIONE