Le Curve Ellittiche in Crittografia: Analisi Tecnica Approfondita

Introduzione

La crittografia a chiave pubblica rappresenta una delle colonne portanti della sicurezza informatica moderna, consentendo comunicazioni sicure e autenticazione digitale in un mondo interconnesso. Con l'aumentare della potenza computazionale e l'emergere di nuove minacce, la ricerca di sistemi crittografici che offrano un elevato livello di sicurezza con la massima efficienza è diventata imperativa. In questo contesto, la crittografia basata sulle curve ellittiche (Elliptic Curve Cryptography - ECC) si è affermata come una delle alternative più potenti e performanti rispetto ai sistemi tradizionali come RSA.

Le curve ellittiche sono oggetti matematici affascinanti, definiti da semplici equazioni cubiche, che possiedono una ricca e profonda struttura algebrica. La loro peculiarità risiede nella possibilità di definire un'operazione di "somma" tra i punti della curva, che dota l'insieme di questi punti della struttura di un gruppo abeliano. È proprio questa struttura di gruppo, unita alla difficoltà computazionale di un problema analogo al logaritmo discreto, a costituire il fondamento della sicurezza dell'ECC.

L'obiettivo di questo documento è fornire una trattazione tecnica esaustiva dei principi matematici e delle applicazioni crittografiche delle curve ellittiche. Si partirà dai fondamenti algebrici, come la teoria dei campi, per poi definire formalmente le curve ellittiche e la loro operazione di gruppo. Successivamente, l'analisi si sposterà sui campi finiti, che costituiscono l'ambiente operativo della crittografia pratica. Verrà esaminato in dettaglio il Problema del Logaritmo Discreto su Curve Ellittiche (ECDLP), la sua intrattabilità computazionale e il suo ruolo come funzione *one-way*. Infine, verranno illustrate le principali applicazioni crittografiche, come lo scambio di chiavi Diffie-Hellman (ECDH) e i sistemi di cifratura, analizzando i protocolli e i parametri di sicurezza.

Capitolo 1: Fondamenti Matematici delle Curve Ellittiche

1.1 Campi Matematici

Per comprendere appieno la struttura delle curve ellittiche, è indispensabile introdurre il concetto algebrico di **campo**. Un campo è l'ambiente matematico in cui le coordinate dei punti di una curva e i coefficienti della sua equazione sono definiti. Prima di definire un campo, è necessario introdurre una struttura più fondamentale: il gruppo.

Strutture Algebriche Fondamentali: Il Gruppo

Un **gruppo** è costituito da un insieme non vuoto G e un'unica operazione binaria che indicheremo genericamente con $*$ che soddisfa quattro proprietà fondamentali, note come assiomi di gruppo:

1. **Chiusura:** Per ogni coppia di elementi $a, b \in G$, il risultato dell'operazione $a * b$ è anch'esso un elemento di G.
2. **Associatività:** Per ogni $a, b, c \in G$, vale l'equazione $(a * b) * c = a * (b * c)$. L'ordine di esecuzione delle operazioni non influisce sul risultato finale.
3. **Esistenza dell'Elemento Neutro:** Esiste un elemento unico $e \in G$, chiamato elemento neutro, tale che per ogni $a \in G$, si ha $a * e = e * a = a$.
4. **Esistenza dell'Inverso:** Per ogni elemento $a \in G$, esiste un elemento unico $a^{-1} \in G$, chiamato inverso di a, tale che $a * a^{-1} = a^{-1} * a = e$.

Se un gruppo soddisfa anche un quinto assioma, quello della commutatività, prende il nome di **gruppo abeliano** (o commutativo).

5. **Commutatività:** Per ogni $a, b \in G$, vale l'equazione $a * b = b * a$. L'ordine degli operandi non influisce sul risultato.

Definizione Formale di Campo

Possiamo ora definire formalmente un campo. Un **campo** è una struttura algebrica costituita da un insieme non vuoto K e due operazioni binarie, chiamate addizione $(+)$ e moltiplicazione $(\cdot)$, che soddisfano i seguenti assiomi:

1. **Struttura di Gruppo Abeliano rispetto all'Addizione:** L'insieme K con l'operazione di addizione, denotato come $(K, +)$, forma un **gruppo abeliano**. L'elemento neutro è 0 e l'inverso di a è $-a$.
2. **Struttura di Gruppo Abeliano rispetto alla Moltiplicazione (escluso lo zero):** L'insieme $K \setminus \{0\}$ (tutti gli elementi di K eccetto l'elemento neutro additivo) con l'operazione di moltiplicazione, denotato come $(K \setminus \{0\}, \cdot)$, forma un **gruppo abeliano**. L'elemento neutro è 1 e l'inverso di a è a^{-1}.
3. **Proprietà Distributiva:** La moltiplicazione è distributiva rispetto all'addizione. Per ogni $a, b, c \in K$, vale $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$.

In sintesi, un campo è un insieme in cui è possibile eseguire addizioni, sottrazioni, moltiplicazioni e divisioni (eccetto la divisione per zero) con le consuete proprietà algebriche.

Esempi di Campi

- $(\mathbb{Q}, +, \cdot)$**:** Il campo dei numeri **razionali**.

- $(\mathbb{R}, +, \cdot)$**:** Il campo dei numeri **reali**.
- $(\mathbb{C}, +, \cdot)$**:** Il campo dei numeri **complessi**.
- $(\mathbb{Z}_p, +, \cdot)$**:** Il campo dei numeri **interi modulo un numero primo** p. Questo è un esempio di **campo finito**, fondamentale per la crittografia. L'insieme è $\{0, 1, 2, \ldots, p-1\}$ e le operazioni sono eseguite modulo p. L'esistenza dell'inverso moltiplicativo per ogni elemento non nullo è garantita dal fatto che p è primo.

L'insieme dei numeri interi $(\mathbb{Z}, +, \cdot)$ **non** è un campo. Pur formando un gruppo abeliano rispetto all'addizione, non soddisfa l'assioma dell'esistenza dell'inverso moltiplicativo. Ad esempio, per $2 \in \mathbb{Z}$, non esiste alcun intero z tale che $2 \cdot z = 1$.

Caratteristica di un Campo

La **caratteristica** di un campo K, denotata con $char(K)$, è definita come il più piccolo intero positivo k tale che la somma dell'elemento neutro moltiplicativo 1 con se stesso per k volte dia come risultato l'elemento neutro additivo 0:

$$\underbrace{1 + 1 + \cdots + 1}_{k \text{ volte}} = k \cdot 1 = 0$$

Se un tale intero positivo k non esiste, la caratteristica del campo è definita come 0.

- $char(\mathbb{Q}) = char(\mathbb{R}) = char(\mathbb{C}) = 0$.
- $char(\mathbb{Z}_p) = p$, poiché in aritmetica modulare p, la somma di p uni è congrua a p, che è congruo a $0 \pmod p$.

La caratteristica di un campo, se non è zero, è sempre un numero primo.

Dimostrazione della Primalità della Caratteristica

Dimostriamo per assurdo perché la caratteristica k di un campo K, se $k \neq 0$, deve essere un numero primo.

1. **Ipotesi per Assurdo:** Assumiamo che k sia la caratteristica non nulla del campo K e che k **non** sia un numero primo. Per definizione, k è il più piccolo intero positivo tale che $k \cdot 1 = 0$.
2. **Decomposizione:** Se k non è primo, allora è un numero composto. Ciò significa che può essere scritto come il prodotto di due interi m e n, tali che $1 < m < k$ e $1 < n < k$. Quindi, $k = m \cdot n$.
3. **Applicazione alla Definizione di Caratteristica:** Sostituiamo $k = m \cdot n$ nell'equazione della caratteristica:

$$(m \cdot n) \cdot 1 = 0$$

4. **Uso delle Proprietà del Campo:** In un campo, la moltiplicazione e l'addizione sono interconnesse dalla distributività, da cui discende che $(m \cdot n) \cdot 1 = (m \cdot 1) \cdot (n \cdot 1)$. Quindi, abbiamo:

$$(m \cdot 1) \cdot (n \cdot 1) = 0$$

5. **Assenza di Divisori dello Zero:** Una proprietà fondamentale di ogni campo è l'assenza di "divisori dello zero". Questo significa che se il prodotto di due elementi è zero, allora almeno uno dei due elementi deve essere zero. Formalmente, se $a \cdot b = 0$, allora $a = 0$ o $b = 0$.
   Applicando questa proprietà alla nostra equazione, deve essere vero che:

$$m \cdot 1 = 0 \quad \text{oppure} \quad n \cdot 1 = 0$$

6. **La Contraddizione:** Ricordiamo che avevamo posto $1 < m < k$ e $1 < n < k$. Se $m \cdot 1 = 0$, abbiamo trovato un intero positivo m che è *strettamente minore* di k e che annulla la somma degli uni. Questo contraddice direttamente la nostra definizione iniziale di k come il *più piccolo* intero positivo con questa proprietà. Lo stesso vale se $n \cdot 1 = 0$.
7. **Conclusione:** L'ipotesi iniziale che k sia un numero composto deve essere falsa. Pertanto, la caratteristica k di un campo, se non è zero, deve necessariamente essere un numero primo.

1.2 Definizione e Geometria delle Curve Ellittiche

Una curva ellittica non è un'ellisse, come il suon nome potrebbe suggerire, ma una curva cubica piana non singolare. la sua nomenclatura deriva storicamente dalla sua relazione con gli integrali ellittici.

L'Equazione Generale di Weierstrass

Una curva ellittica E definita su un campo K è l'insieme dei punti $(x, y) \in K \times K$ che soddisfano l'**equazione generale di Weierstrass**:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

dove i coefficienti a_1, a_2, a_3, a_4, a_6 sono elementi del campo K. A questo insieme di punti si aggiunge un punto speciale, chiamato **punto all'infinito**, denotato con O.

La Forma Normale di Weierstrass

Se la caratteristica del campo K è diversa da 2 e 3 ($char(K) \neq 2, 3$), è possibile semplificare l'equazione generale attraverso un cambio di variabili ammissibile. Questa semplificazione porta alla **forma normale di Weierstrass**:

$$y^2 = x^3 + ax + b$$

dove $a, b \in K$.

Questa è la forma che verrà utilizzata nel resto del documento, poiché i campi utilizzati in crittografia (come $\mathbb{Z}_p$ per $p > 3$) soddisfano questa condizione.

Il Punto all'Infinito (O)

Per dotare l'insieme dei punti di una curva ellittica di una struttura di gruppo algebricamente coerente, è necessario estenderlo dal piano affine K^2 a un contesto geometrico più completo: il **piano proiettivo** $\mathbb{P}^2(K)$.

Il piano proiettivo $\mathbb{P}^2(K)$ può essere concepito come il piano affine K^2 a cui viene aggiunta una "retta all'infinito", sulla quale si incontrano le rette parallele. Formalmente, i suoi punti sono rappresentati da **coordinate omogenee** $(X : Y : Z)$, che sono terne di elementi di K (non tutti nulli) definite a meno di un fattore di scala non nullo. Ciò significa che il punto $(X : Y : Z)$ è identico al punto $(\lambda X : \lambda Y : \lambda Z)$ per qualsiasi $\lambda \in K \setminus \{0\}$. La connessione con il piano affine si stabilisce ponendo $x = X/Z$ e $y = Y/Z$ per tutti i punti con $Z \neq 0$. I punti con $Z = 0$ costituiscono la retta all'infinito.

Il motivo per cui si **omogeneizza l'equazione** è per renderla compatibile con questa nuova struttura di coordinate. Un'equazione come $y^2 = x^3 + ax + b$ non è ben definita in coordinate omogenee, poiché il suo soddisfacimento dipenderebbe dal fattore di scala scelto. Sostituendo $x = X/Z$ e $y = Y/Z$ e moltiplicando per la potenza di Z necessaria a eliminare i denominatori, si ottiene l'equazione omogenea $Y^2Z = X^3 + aXZ^2 + bZ^3$. Questa nuova equazione ha la proprietà cruciale che tutti i suoi termini hanno lo stesso grado totale che nel nostro caso si attesta a 3. Ciò garantisce che se la terna (X, Y, Z) la soddisfa, allora anche qualsiasi sua multipla $(\lambda X, \lambda Y, \lambda Z)$ la soddisferà, rendendo il concetto di "punto sulla curva" matematicamente consistente nel piano proiettivo.

È proprio questo processo formale che rivela l'esistenza di un unico punto aggiuntivo sulla curva: il **punto all'infinito**, O. Imponendo la condizione $Z = 0$ nell'equazione omogenea, si trova che l'unica soluzione è $(0 : Y : 0)$, che per convenzione si normalizza a $(0 : 1 : 0)$. Questo punto, assente nel piano affine, è essenziale per la struttura algebrica della curva, agendo da **elemento neutro** del gruppo e garantendo che le operazioni siano sempre definite.

Da un punto di vista geometrico, O può essere interpretato in due modi complementari:

1. **Come Intersezione di Rette Verticali:** È il punto di intersezione comune a **tutte le rette verticali** del piano. Ogni retta della forma $x = c$ interseca la curva in due punti affini (c, y) e $(c, -y)$ (se esistono nel campo K) e nel punto all'infinito O.

2. **Come Chiusura della Curva:** È fondamentale comprendere che O è un punto **singolo**. Le due "braccia" della curva che si estendono indefinitamente per valori di y positivi e negativi si incontrano in questo unico punto. Questa proprietà di unicità è ciò che "chiude" la curva, ma descriverla come un "anello" è un'analogia topologica che può essere fuorviante. La sua funzione primaria in questo contesto non è topologica, ma algebrica: agire come **elemento neutro** del gruppo, garantendo che le operazioni siano sempre definite.

L'inclusione del punto O ha implicazioni strutturali cruciali:

1. **Esistenza dell'Elemento Neutro:** O funge da **elemento neutro** per l'operazione di addizione, soddisfacendo l'assioma fondamentale $P + O = P$ per ogni punto P sulla curva.
2. **Chiusura del Gruppo:** O garantisce la **chiusura** dell'operazione, risolvendo casi che nel piano affine sarebbero indefiniti. L'esempio cruciale è la somma di un punto $P = (x, y)$ con il suo inverso $-P = (x, -y)$. La retta verticale che li unisce interseca la curva proprio in O, portando alla relazione fondamentale $P + (-P) = O$. Senza O, l'operazione non sarebbe definita per ogni coppia di punti, e la struttura di gruppo verrebbe meno.
3. **Coerenza Algebrica:** La sua presenza assicura che la legge di gruppo sia **sempre ben definita**. Ad esempio, se la tangente a un punto P è verticale ovvero ha equanzione $y_P = 0$, l'operazione di raddoppio $2P$ risulta correttamente in O, evitando singolarità nel calcolo.

La Condizione di Non-Singolarità

Perché una curva definita dall'equazione $y^2 = x^3 + ax + b$ possa essere utilizzata per la crittografia, deve essere **non singolare**. Una curva è **singolare** se possiede punti in cui la tangente non è definita in modo univoco, come nodi o cuspidi.

[IMMAGINE: Esempio di curva ellittica singolare con un nodo e una con una cuspide.]

La condizione algebrica per la non-singolarità è che il discriminante del polinomio cubico in x sia diverso da zero:

$$\Delta = -16(4a^3 + 27b^2) \neq 0$$

Questo è equivalente a richiedere che:

$$4a^3 + 27b^2 \neq 0$$

Questa condizione garantisce che il polinomio $x^3 + ax + b$ non abbia radici multiple, il che è fondamentale per la definizione coerente dell'operazione di somma in tutti i punti della curva.

Geometria sui Numeri Reali

Se il campo di definizione è $\mathbb{R}$, possiamo visualizzare le curve ellittiche nel piano cartesiano. L'insieme dei punti è $E(a,b) = \{(x,y) \in \mathbb{R}^2 : y^2 = x^3 + ax + b\} \cup \{O\}$.
La forma della curva dipende dalle radici del polinomio $x^3 + ax + b$:

- **Tre radici reali distinte:** La curva ha due componenti sconnesse.
- **Una radice reale:** La curva ha una singola componente connessa.

[IMMAGINE: Grafico di una curva ellittica sui reali con una componente, ad esempio $y^2 = x^3 - x + 1$.]
[IMMAGINE: Grafico di una curva ellittica sui reali con due componenti, ad esempio $y^2 = x^3 - 3x + 1$.]

Tutte le curve ellittiche non singolari sono simmetriche rispetto all'asse delle ascisse, poiché se (x,y) è un punto della curva, anche $(x,-y)$ lo è, dato che y appare al quadrato nell'equazione.

Intersezione tra Curve Ellittiche e Rette

Una proprietà geometrica fondamentale, che costituisce la base della legge di gruppo, è che ogni retta interseca una curva ellittica in **esattamente tre punti**, a condizione che questi punti siano contati nel piano proiettivo e con la loro corretta **molteplicità**.

L'analisi si fonda sull'intersezione algebrica. Per una retta non verticale di equazione $y = \lambda x + \nu$, la sostituzione nell'equazione della curva $y^2 = x^3 + ax + b$ produce un'equazione polinomiale di terzo grado in x:

$$x^3 - \lambda^2 x^2 + (a - 2\lambda\nu)x + (b - \nu^2) = 0$$

Il Teorema Fondamentale dell'Algebra garantisce che questa equazione abbia sempre tre radici assumendo di trovarsi in un campo algebricamente chiuso come $\mathbb{C}$ e contate con molteplicità. Questa proprietà algebrica si traduce geometricamente nei seguenti scenari:

- **Tre intersezioni distinte:** L'equazione ha tre radici distinte, corrispondenti a tre punti $(x_1,y_1), (x_2,y_2), (x_3,y_3)$ distinti sulla curva.
- **Un punto di tangenza:** Se la retta è tangente alla curva in un punto P, l'equazione ha una radice doppia. Tale punto P conta come **due** intersezioni ovvero presenterà una molteplicità pari a 2 e la retta interseca la curva in un terzo punto distinto R.
- **Un punto di flesso:** In casi particolari, una retta può intersecare la curva in un unico punto P con molteplicità 3. Ciò corrisponde a una radice tripla dell'equazione.

Il caso di una retta verticale $x = c$ si adatta perfettamente a questa regola nel piano proiettivo: essa interseca la curva nei due punti affini (c,y) e $(c,-y)$ e nel **punto all'infinito** O, totalizzando anche in

questo caso tre intersezioni.

Questa invarianza, garantita da un principio più generale noto come **Teorema di Bézout**, è ciò che rende la legge di gruppo robusta. Assicura che, dati due punti P e Q, la retta che li congiunge determinerà *sempre* un terzo punto di intersezione. Ciò ci permette di definire la somma $P + Q$ in modo universale e senza eccezioni.

Capitolo 2: L'Algebra delle Curve Ellittiche: L'Operazione di Gruppo

Il motivo principale per cui le curve ellittiche sono così potenti in crittografia è che l'insieme dei loro punti, $E(K)$, forma un **gruppo abeliano** rispetto a un'operazione di addizione definita geometricamente.

2.1 La Struttura di Gruppo Abeliano

Un gruppo abeliano è un insieme dotato di un'operazione binaria che soddisfa le proprietà di chiusura, associatività, commutatività, esistenza dell'elemento neutro e dell'inverso per ogni elemento. Vedremo come l'operazione di "somma" tra punti di una curva ellittica soddisfi tutti questi requisiti. L'elemento neutro del gruppo è il punto all'infinito O.

2.2 L'Operazione di Addizione di Punti

L'operazione di addizione è definita a partire da una regola geometrica basata sulla collinearità.

Definizione Geometrica

La regola fondamentale è: **Se tre punti P, Q, R di una curva ellittica sono allineati, la loro somma è l'elemento neutro O**:

$$P + Q + R = O$$

Da questa regola, possiamo derivare come sommare due punti qualsiasi P e Q per ottenere un terzo punto $S = P + Q$.

1. **Tracciare la retta:** Si traccia una retta passante per i punti P e Q.
2. **Trovare il terzo punto:** Per la proprietà vista in precedenza, questa retta intersecherà la curva in un terzo punto, che chiamiamo R. (Se $P = Q$, la retta è la tangente alla curva in P).

3. **Definire la somma:** La somma $P + Q$ non è R, ma il suo **riflesso rispetto all'asse delle** x. Chiamiamo questo punto $S = -R$.

Quindi, $P + Q = -R$. Sostituendo nella regola fondamentale, abbiamo $P + Q + (-S) = O$, che è coerente.

[IMMAGINE: Illustrazione della somma di due punti distinti P e Q su una curva ellittica. La retta per P e Q interseca la curva in R. La somma $P + Q$ è il punto S, riflesso di R rispetto all'asse x.]
[IMMAGINE: Illustrazione del raddoppio di un punto P. La tangente in P interseca la curva in R. La somma $2P$ è il punto S, riflesso di R rispetto all'asse x.]

Derivazione delle Formule Algebriche

Vediamo ora come tradurre questa costruzione geometrica in formule algebriche per una curva $y^2 = x^3 + ax + b$.

Caso 1: Addizione di due punti distinti ($P \neq Q$)

Siano $P = (x_P, y_P)$ e $Q = (x_Q, y_Q)$. Poiché $P \neq Q$, se $x_P = x_Q$, allora $y_P = -y_Q$. In questo caso, la retta che li congiunge è verticale e interseca la curva nel punto all'infinito. Quindi $P + Q = O$.

Se $x_P \neq x_Q$, la retta passante per P e Q ha equazione $y = \lambda x + \nu$, dove il coefficiente angolare λ è:

$$\lambda = \frac{y_Q - y_P}{x_Q - x_P}$$

Sostituendo y nell'equazione della curva, otteniamo l'equazione cubica per le ascisse dei punti di intersezione:

$$(\lambda x + \nu)^2 = x^3 + ax + b$$

$$x^3 - \lambda^2 x^2 + \cdots = 0$$

Le radici di questo polinomio sono x_P, x_Q, x_R, dove $R = (x_R, y_R)$ è il terzo punto di intersezione. Per le formule di Viète, la somma delle radici di un polinomio monico di grado 3 è uguale al coefficiente del termine di secondo grado cambiato di segno. Pertanto:

$$x_P + x_Q + x_R = \lambda^2$$

Da cui possiamo ricavare l'ascissa di R:

$$x_R = \lambda^2 - x_P - x_Q$$

L'ordinata y_R si trova sulla retta:

$$y_R = \lambda(x_R - x_P) + y_P$$

La somma $S = P + Q$ è il punto $-R$. Quindi, $S = (x_S, y_S) = (x_R, -y_R)$. Le coordinate di $S = P + Q$ sono:

$$x_S = \lambda^2 - x_P - x_Q$$

$$y_S = -(y_P + \lambda(x_S - x_P)) = \lambda(x_P - x_S) - y_P$$

Caso 2: Raddoppio di un punto ($P = Q$)

Se $P = Q$, la retta da considerare è la tangente alla curva in $P = (x_P, y_P)$. Per trovare il coefficiente angolare λ, deriviamo implicitamente l'equazione della curva rispetto a x:

$$2y\frac{dy}{dx} = 3x^2 + a$$

$$\lambda = \frac{dy}{dx} = \frac{3x_P^2 + a}{2y_P}$$

Questa formula è valida solo se $y_P \neq 0$. Se $y_P = 0$, la tangente è verticale e interseca la curva nel punto all'infinito. In questo caso, $2P = O$.

Se $y_P \neq 0$, le formule sono analoghe al caso precedente, ma con $x_Q = x_P$ e $y_Q = y_P$:

$$x_R = \lambda^2 - 2x_P$$

E di nuovo, $S = 2P = (x_S, y_S) = (x_R, -y_R)$. Le coordinate di $S = 2P$ sono:

$$x_S = \left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P$$

$$y_S = \left(\frac{3x_P^2 + a}{2y_P}\right)(x_P - x_S) - y_P$$

Proprietà di Gruppo

- **Chiusura:** Le formule algebriche mostrano che la somma di due punti sulla curva produce sempre un altro punto le cui coordinate soddisfano l'equazione della curva.
- **Elemento Neutro:** Il punto all'infinito O agisce da elemento neutro. Per ogni punto P, $P + O = P$. Geometricamente, la retta che passa per P e O è una retta verticale che interseca la curva in P e $-P$. Quindi il terzo punto di intersezione è $-P$. Il suo riflesso è $-(-P) = P$.
- **Inverso:** Per ogni punto $P = (x, y)$, il suo inverso additivo è $-P = (x, -y)$. La retta che li congiunge è verticale, e il terzo punto di intersezione è O. Il riflesso di O è O stesso, quindi $P + (-P) = O$.
- **Commutatività:** $P + Q = Q + P$. Questa proprietà è evidente, poiché la retta che passa per P e Q è la stessa che passa per Q e P.
- **Associatività:** $(P + Q) + R = P + (Q + R)$. Questa è la proprietà più complessa da dimostrare algebricamente. La sua dimostrazione rigorosa esula dagli scopi di questo documento, ma è una proprietà fondamentale che garantisce la coerenza della struttura di gruppo.

2.3 Moltiplicazione Scalare di un Punto (kP)

La **moltiplicazione scalare** è l'operazione di sommare un punto P a se stesso un numero intero k di volte:

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

Questa operazione è l'analogo della potenza nell'aritmetica modulare ed è centrale in crittografia. Ad esempio, $3P = P + P + P = (2P) + P$.

Calcolare kP in modo naïf ovvero sommando P per $k - 1$ volte sarebbe inefficiente per valori grandi di k. Si utilizza invece un algoritmo molto più rapido detto **Algoritmo Double-and-Add**.

Algoritmo Double-and-Add

Questo algoritmo permette di calcolare kP in tempo logaritmico rispetto a k. Si basa sulla rappresentazione binaria di k.

Sia $k = (b_m b_{m-1} \dots b_1 b_0)_2$ la rappresentazione binaria di k, dove $b_m = 1$.
L'algoritmo funziona come segue:

1. **Inizializzazione:** Si pone $Q = P$.
2. **Iterazione:** Si scorrono i bit di k da sinistra a destra, da $m-1$ fino a 0:
   - **Double:** Si raddoppia il punto corrente: $Q = 2Q$.
   - **Add (se necessario):** Se il bit corrente b_i è 1, si aggiunge P: $Q = Q + P$.
3. **Risultato:** Il valore finale di Q è kP.

Esempio: Calcolo di $13P$
La rappresentazione binaria di 13 è 1101_2. ($b_3 = 1, b_2 = 1, b_1 = 0, b_0 = 1$).

1. **Inizio (bit $b_3 = 1$):** $Q = P$.
2. **Iterazione per $b_2 = 1$:**
   - Double: $Q = 2Q = 2P$.
   - Add (poiché $b_2 = 1$): $Q = Q + P = 2P + P = 3P$.
3. **Iterazione per $b_1 = 0$:**
   - Double: $Q = 2Q = 2(3P) = 6P$.
   - Add (poiché $b_1 = 0$): Nessuna aggiunta. Q rimane $6P$.
4. **Iterazione per $b_0 = 1$:**
   - Double: $Q = 2Q = 2(6P) = 12P$.
   - Add (poiché $b_0 = 1$): $Q = Q + P = 12P + P = 13P$.

Il risultato finale è $13P$. La complessità di questo algoritmo è di circa $\log_2(k)$ raddoppi e $\log_2(k)/2$ somme in media, rendendolo estremamente efficiente.

Capitolo 3: Curve Ellittiche su Campi Finiti: Il Cuore della Crittografia

3.1 Motivazioni per l'Uso di Campi Finiti

Le curve ellittiche definite sui numeri reali sono eccellenti per la visualizzazione e la comprensione geometrica, ma sono inadatte per la crittografia per due ragioni principali:

1. **Errori di arrotondamento:** L'aritmetica in $\mathbb{R}$ su di un calcolatore è approssimata tramite numeri in virgola mobile. Le operazioni crittografiche richiedono una precisione assoluta; qualsiasi errore di arrotondamento distruggerebbe l'integrità dei calcoli, rendendo di fatto inaffidabile il processo.
2. **Insieme infinito:** L'insieme dei punti è infinito e continuo, il che rende difficile definire problemi computazionalmente "difficili" in modo discreto.

I **campi finiti**, come $\mathbb{Z}_p$, risolvono entrambi i problemi. L'aritmetica modulare è esatta e opera su un insieme finito di elementi. Questo ambiente discreto e finito è ideale per costruire problemi con caso pessimo di risoluzione computazionalmente esponenziale, che sono il fondamento della sicurezza crittografica.

3.2 Curve Ellittiche su $\mathbb{Z}_p$

Una curva ellittica su un campo finito primo $\mathbb{Z}_p$ con $p > 3$ e primo è definita in modo analogo al caso reale, ma tutte le operazioni sono eseguite in modulo p.

Definizione Formale

Dati $a, b \in \mathbb{Z}_p$ tali che $4a^3 + 27b^2 \not\equiv 0 \pmod p$, la curva ellittica $E_p(a, b)$ è l'insieme dei punti (x, y) con $x, y \in \mathbb{Z}_p$ che soddisfano l'equazione:

$$y^2 \equiv x^3 + ax + b \pmod p$$

insieme al punto all'infinito O.

$$E_p(a, b) = \{(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p : y^2 \equiv x^3 + ax + b \pmod p\} \cup \{O\}$$

L'insieme dei punti non è più una curva continua, ma una nuvola di punti discreti le cui coordinate sono intere comprese tra 0 e $p - 1$.

[IMMAGINE: Grafico a dispersione dei punti di una curva ellittica su un campo finito, ad esempio $E_{97}(2, 3)$, che mostra la distribuzione apparentemente casuale ma simmetrica dei punti.]

Inversi Modulari e Algoritmo di Euclide Esteso

Le formule per l'addizione dei punti coinvolgono divisioni, come nel calcolo di λ. In un campo finito $\mathbb{Z}_p$, la divisione per un numero d è definita come la moltiplicazione per il suo **inverso moltiplicativo modulare**, $d^{-1} \pmod p$.
L'inverso moltiplicativo di un intero $d \in \mathbb{Z}_p \setminus \{0\}$ è un intero d^{-1} tale che:

$$d \cdot d^{-1} \equiv 1 \pmod{p}$$

Questo inverso esiste ed è unico se e solo se $\text{MCD}(d, p) = 1$. Poiché p è primo e $d \in \{1, \ldots, p-1\}$, questa condizione è sempre soddisfatta.

Per calcolare l'inverso modulare si utilizza l'**Algoritmo di Euclide Esteso**. Questo algoritmo, dato due interi d e p, trova due interi s e t tali che:

$$s \cdot d + t \cdot p = \text{MCD}(d, p)$$

Se $\text{MCD}(d, p) = 1$, allora abbiamo $s \cdot d + t \cdot p = 1$. Considerando questa equazione modulo p, il termine $t \cdot p$ diventa zero:

$$s \cdot d \equiv 1 \pmod{p}$$

Quindi, $s \pmod{p}$ è l'inverso moltiplicativo di d modulo p.

Esempio: Calcolare $7^{-1} \pmod{23}$.
Applichiamo l'algoritmo di Euclide Esteso a 23 e 7:

- $23 = 3 \cdot 7 + 2$
- $7 = 3 \cdot 2 + 1$
- $2 = 2 \cdot 1 + 0 \implies \text{MCD}(23, 7) = 1.$

Ora lavoriamo a ritroso per esprimere 1 come combinazione lineare di 7 e 23:

- $1 = 7 - 3 \cdot 2$
- Dalla prima riga, sappiamo che $2 = 23 - 3 \cdot 7$. Sostituiamo questo nella riga sopra:
- $1 = 7 - 3 \cdot (23 - 3 \cdot 7)$
- $1 = 7 - 3 \cdot 23 + 9 \cdot 7$
- $1 = 10 \cdot 7 - 3 \cdot 23$

Considerando questa equazione modulo 23:
$10 \cdot 7 \equiv 1 \pmod{23}$.
Quindi, $7^{-1} \equiv 10 \pmod{23}$.

Esempio Dettagliato di Addizione in $\mathbb{Z}_p$

Consideriamo la curva E definita dall'equazione $y^2 \equiv x^3 + x + 1 \pmod{23}$.
Scegliamo due punti sulla curva: $P = (1, 7)$ e $Q = (3, 10)$.

Passo 1: Verifica dell'appartenenza dei punti alla curva

Prima di procedere, verifichiamo che i punti scelti soddisfino effettivamente l'equazione della curva.

- **Per il punto $P = (1, 7)$:**
  - Lato sinistro: $y_P^2 = 7^2 = 49 \equiv 3 \pmod{23}$.
  - Lato destro: $x_P^3 + x_P + 1 = 1^3 + 1 + 1 = 3 \pmod{23}$.
  - Poiché $3 \equiv 3$, il punto P appartiene alla curva.
- **Per il punto $Q = (3, 10)$:**
  - Lato sinistro: $y_Q^2 = 10^2 = 100 \equiv 8 \pmod{23}$ (poiché $100 = 4 \cdot 23 + 8$).
  - Lato destro: $x_Q^3 + x_Q + 1 = 3^3 + 3 + 1 = 27 + 4 = 31 \equiv 8 \pmod{23}$.
  - Poiché $8 \equiv 8$, il punto Q appartiene alla curva.

Passo 2: Calcolo del coefficiente angolare λ

Utilizziamo la formula per l'addizione di punti distinti:

$$\lambda \equiv \frac{y_Q - y_P}{x_Q - x_P} \equiv (y_Q - y_P)(x_Q - x_P)^{-1} \pmod{23}$$

Sostituendo i valori:

$$\lambda \equiv (10 - 7)(3 - 1)^{-1} \equiv 3 \cdot 2^{-1} \pmod{23}$$

Ora dobbiamo calcolare l'inverso moltiplicativo di 2 modulo 23. Cerchiamo un numero k tale che $2 \cdot k \equiv 1 \pmod{23}$. In questo caso semplice, possiamo vedere che $2 \cdot 12 = 24 \equiv 1 \pmod{23}$. Quindi, $2^{-1} \equiv 12 \pmod{23}$.

Sostituiamo l'inverso per trovare λ:

$$\lambda \equiv 3 \cdot 12 = 36 \equiv 13 \pmod{23}$$

Passo 3: Calcolo delle coordinate del punto somma $S = P + Q$

Ora usiamo le formule per le coordinate di $S = (x_S, y_S)$:

- **Calcolo dell'ascissa x_S:**

$$x_S \equiv \lambda^2 - x_P - x_Q \pmod{23}$$

$$x_S \equiv 13^2 - 1 - 3 = 169 - 4 = 165 \pmod{23}$$

Per ridurre 165 modulo 23, calcoliamo $165 = 7 \cdot 23 + 4$. Quindi:

$$x_S \equiv 4 \pmod{23}$$

- **Calcolo dell'ordinata y_S:**

$$y_S \equiv \lambda(x_P - x_S) - y_P \pmod{23}$$

$$y_S \equiv 13(1 - 4) - 7 = 13(-3) - 7 = -39 - 7 = -46 \pmod{23}$$

Per ridurre -46 modulo 23, aggiungiamo multipli di 23 fino a ottenere un numero positivo: $-46 + 2 \cdot 23 = -46 + 46 = 0$. Quindi:

$$y_S \equiv 0 \pmod{23}$$

Passo 4: Risultato finale
La somma dei punti $P = (1, 7)$ e $Q = (3, 10)$ sulla curva data è il punto $S = (4, 0)$.

Passo 5 (Opzionale): Verifica del punto risultante
Per confermare la correttezza del calcolo, verifichiamo che il punto $S = (4, 0)$ giaccia anch'esso sulla curva:

- Lato sinistro: $y_S^2 = 0^2 = 0 \pmod{23}$.
- Lato destro: $x_S^3 + x_S + 1 = 4^3 + 4 + 1 = 64 + 5 = 69 \pmod{23}$.
- Poiché $69 = 3 \cdot 23$, abbiamo $69 \equiv 0 \pmod{23}$.
  Il punto S appartiene alla curva, confermando la proprietà di chiusura del gruppo e la correttezza dei nostri calcoli.

Ordine della Curva e Ordine del Punto

La sicurezza della crittografia su curve ellittiche non deriva semplicemente dall'esistenza della struttura di gruppo, ma dalla sua **dimensione** e dalla sua **struttura interna**. I concetti di *ordine della curva* e *ordine di un punto* sono quindi fondamentali, poiché definiscono la grandezza del campo di gioco per un crittanalista e determinano la reale difficoltà del Problema del Logaritmo Discreto su Curve Ellittiche (ECDLP).

Ordine della Curva e il Teorema di Hasse: Quantificare lo Spazio delle Soluzioni

L'**ordine** di una curva ellittica $E_p(a, b)$, denotato $|E_p(a, b)|$, è il numero totale di punti discreti che soddisfano la sua equazione sul campo $\mathbb{Z}_p$, incluso il punto all'infinito O. Questo valore rappresenta la cardinalità del gruppo $(E_p(a, b), +)$.

A prima vista, potrebbe non essere ovvio quanti punti ci si possa aspettare su una data curva. Il **Teorema di Hasse** fornisce una stima straordinariamente precisa e potente, legando l'ordine della curva alla dimensione del campo sottostante:

$$p + 1 - 2\sqrt{p} \leq |E_p(a, b)| \leq p + 1 + 2\sqrt{p}$$

Implicazioni Crittografiche del Teorema di Hasse:
Il teorema ci dice che l'ordine di una curva è sempre "vicino" alla dimensione del campo $p + 1$. Per la crittografia, questo è un risultato cruciale:

1. **Garanzia di Grandezza:** Scegliendo un numero primo p sufficientemente grande (es. 256 bit), il Teorema di Hasse ci garantisce che l'ordine della curva sarà anch'esso un numero molto grande, dello stesso ordine di grandezza di p. Questo assicura che lo spazio totale dei punti sia vasto, rendendo impraticabili attacchi di forza bruta che consistono nell'enumerare tutti i punti.
2. **Prevedibilità:** Fornisce un intervallo entro cui cercare l'ordine esatto della curva, un passo fondamentale nella selezione di curve sicure (eseguito con algoritmi di conteggio dei punti come l'algoritmo di Schoof).

Ordine di un Punto e Sottogruppi Ciclici: Il Vero Campo di Gioco Crittografico

Mentre l'ordine della curva è importante, la sicurezza pratica si basa su un concetto più specifico: l'**ordine di un punto**. L'ordine di un punto $P \in E_p(a, b)$ è il più piccolo intero positivo n tale che:

$$nP = \underbrace{P + P + \cdots + P}_{n \text{ volte}} = O$$

Il punto P genera un **sottogruppo ciclico** di $E_p(a, b)$, denotato come $\langle P \rangle$, che è l'insieme di tutti i multipli di P:

$$\langle P \rangle = \{O, P, 2P, 3P, \ldots, (n-1)P\}$$

Questo sottogruppo ha esattamente n elementi. Il **Teorema di Lagrange**, un risultato fondamentale della teoria dei gruppi, afferma che l'ordine di qualsiasi sottogruppo deve essere un divisore dell'ordine del gruppo principale. Di conseguenza, l'ordine n di qualsiasi punto P deve essere un divisore dell'ordine della curva $|E_p(a,b)|$. È fondamentale capire che i protocolli crittografici come ECDH e ECDSA **non operano sull'intero gruppo** $E_p(a,b)$, ma all'interno di un **sottogruppo ciclico generato da un punto base pubblico** B. La sicurezza del sistema dipende quindi direttamente dalla dimensione n di questo sottogruppo. Per garantire la massima sicurezza, l'obiettivo è scegliere un punto base B il cui ordine n sia un **numero primo molto grande**. La ragione di questo requisito stringente risiede nella vulnerabilità a un attacco specifico, overo l'**l'algoritmo di Pohlig-Hellman** che approfondiremo in seguito (TODO) Per il momento ci è sufficiente sapere che questo attacco sfrutta la fattorizzazione dell'ordine n. Se n è un numero composto con fattori primi piccoli, ad esempio $n = q_1^{e_1} \cdot q_2^{e_2} \cdot \ldots \cdot q_k^{e_k}$, l'attacco di Pohlig-Hellman permette di scomporre un singolo "grande" problema ECDLP nel gruppo di ordine n in una serie di problemi ECDLP "piccoli" nei sottogruppi di ordine q_i. Questi problemi più piccoli possono essere risolti in modo efficiente e i risultati ricombinati per trovare la soluzione originale.

Esempio Concettuale: Risolvere un ECDLP in un gruppo di ordine $n = 1.000.000$ potrebbe essere proibitivo. Ma se n si fattorizza in $10^6 = 2^6 \cdot 5^6$, l'attacco lo riduce a problemi molto più semplici nei sottogruppi di ordine 2 e 5.

Scegliendo un ordine n che sia un numero primo, la sua unica fattorizzazione è $n = n^1$. L'attacco di Pohlig-Hellman non offre alcun vantaggio e un crittanalista è costretto a utilizzare algoritmi generici (come il Pollard's Rho), la cui complessità dipende dalla radice quadrata di n. Per un n primo di 256 bit, questo è computazionalmente intrattabile.

Il Cofattore

Dato che n, l'ordine del sottogruppo, deve dividere $|E(\mathbb{Z}_p)|$, l'ordine della curva, possiamo definire un intero h, chiamato **cofattore**, come:

$$h = \frac{|E(\mathbb{Z}_p)|}{n}$$

Il cofattore h quantifica il rapporto tra la cardinalità del gruppo totale dei punti e quella del sottogruppo ciclico utilizzato per le operazioni crittografiche. Il suo valore non è un mero dettaglio implementativo, ma un parametro di sicurezza critico la cui selezione influenza direttamente la robustezza di un crittosistema ECC contro specifiche classi di attacchi.

La principale motivazione per un'attenta selezione del cofattore è la mitigazione degli attacchi a sottogruppi di ordine minore o small subgroup attacks. Un avversario può sfruttare un cofattore $h > 1$

con fattori primi di piccole dimensioni per orchestrare un attacco volto a estrarre informazioni sulla chiave privata di una vittima. Per neutralizzare questa minaccia, la progettazione di curve crittografiche sicure segue due approcci principali.

La scelta più sicura e computazionalmente semplice da gestire è un cofattore unitario. Quando $h = 1$, l'ordine della curva $|E(\mathbb{Z}_p)|$ è esso stesso un grande numero primo, e di conseguenza $n = |E(\mathbb{Z}_p)|$. Il sottogruppo di lavoro coincide con l'intero gruppo di punti della curva (escluso O).

Alcune curve ad alte prestazioni, progettate specificamente per massimizzare l'efficienza e la resistenza ad attacchi a canali laterali, utilizzano deliberatamente un cofattore piccolo, ma maggiore di 1. Un cofattore è considerato sicuro se è una piccola potenza di due, tipicamente $h = 4$ o $h = 8$. Un cofattore come $h = 8$ è ritenuto sicuro perché la massima quantità di informazione sulla chiave privata che un attacco a sottogruppi potrebbe teoricamente rivelare è $d_V \pmod 8$, ovvero solo 3 bit, un valore trascurabile per chiavi di 256 bit o più. Il beneficio ottenuto in termini di performance e sicurezza fisica supera ampiamente questo rischio teorico minimo.

Scelta dei Parametri di Dominio

Possiamo dunque dire che la selezione di una curva sicura è un processo di selezione che consiste nel trovare una tupla di **parametri di dominio** (p, a, b, B, n, h), dove:

- p è un primo che definisce il campo.
- a, b definiscono la curva.
- $|E_p(a, b)|$ è l'ordine della curva.
- n è un grande numero primo che divide $|E_p(a, b)|$.
- B è un punto generatore di un sottogruppo di ordine n.
- h è un cofattore piccolo.

Questo insieme di parametri definisce l'ambiente sicuro in cui tutte le operazioni crittografiche avranno luogo.

3.3 Cenni di Curve Ellittiche su Campi Binari $GF(2^m)$

Oltre ai campi primi, un'altra famiglia di campi finiti molto usata in crittografia è quella dei **campi binari** ovvero quei campi che presentano caratteristica 2, denotati $GF(2^m)$ o $\mathbb{F}_{2^m}$. Gli elementi di questo campo sono polinomi di grado al più $m - 1$ con coefficienti in $\mathbb{Z}_2 = \{0, 1\}$. L'aritmetica è polinomiale, modulo un polinomio irriducibile di grado m.
Poiché la caratteristica è 2, la forma normale di Weierstrass non è valida. Si usa un'equazione diversa:

- **Non-supersingolare:** $y^2 + xy = x^3 + ax^2 + b$ (con $b \neq 0$)

- **Supersingolare:** $y^2 + ay = x^3 + bx + c$
  Le formule per l'addizione dei punti sono diverse ma derivate con principi simili. Le curve su campi binari sono state storicamente vantaggiose per le implementazioni hardware, anche se oggi le implementazioni su campi primi sono spesso preferite per la loro maggiore semplicità concettuale e sicurezza contro alcuni attacchi specifici.

Capitolo 4: Il Problema del Logaritmo Discreto su Curve Ellittiche (ECDLP)

La sicurezza di tutti i sistemi crittografici basati su curve ellittiche si fonda sulla difficoltà computazionale di un problema specifico: il Problema del Logaritmo Discreto su Curve Ellittiche (ECDLP). La sua intrattabilità è il pilastro che garantisce la robustezza di protocolli come ECDH ed ECDSA.

4.1 Definizione e Natura dell'ECDLP

Per comprendere l'ECDLP, è utile richiamare il suo analogo nel mondo della moltiplicazione modulare, il Problema del Logaritmo Discreto (DLP) "classico". Definito nel gruppo moltiplicativo di un campo finito $\mathbb{Z}_p^*$, il DLP consiste, dati un generatore g e un elemento h, nel trovare l'intero k tale che $h \equiv g^k \pmod p$.

L'ECDLP è la trasposizione di questo problema nel gruppo additivo dei punti di una curva ellittica.

Definizione Formale dell'ECDLP:
Dati una curva ellittica E su un campo finito, un punto base P sulla curva di ordine n, e un altro punto Q che è un multiplo scalare di P (ovvero $Q \in \langle P \rangle$), il problema consiste nel trovare l'unico intero $k \in \{0, 1, \ldots, n-1\}$ tale che:

$$Q = kP$$

Questo intero k è chiamato il **logaritmo discreto di Q in base P**.

La natura dell'ECDLP ne fa una funzione crittografica ideale, in quanto è una funzione one-way ovvero unidirezionale.

- **Direzione Facile (Moltiplicazione Scalare):** Dati k e P, calcolare il punto $Q = kP$ è computazionalmente efficiente. Utilizzando l'algoritmo Double-and-Add visto in precedenza, questa operazione può essere eseguita in tempo logaritmico, $\mathcal{O}(\log k)$.

- **Direzione Difficile (Logaritmo Discreto):** Dati i punti P e Q, determinare lo scalare k è, per curve e parametri scelti opportunamente, un problema computazionalmente intrattabile.

4.2 Difficoltà Computazionale dell'ECDLP

La robustezza crittografica di un sistema ECC è determinata dalla complessità computazionale richiesta per risolvere l'ECDLP. Questa difficoltà non è assoluta, ma dipende criticamente dalla scelta dei parametri del gruppo, in particolare dalla struttura matematica del suo ordine.

L'Attacco di Pohlig-Hellman: Sfruttare la Struttura dell'Ordine

Come anticipato nel capitolo precedente, la sicurezza dell'ECDLP è legata non solo alla grandezza dell'ordine n del sottogruppo, ma alla sua **struttura aritmetica**. L'attacco di Pohlig-Hellman è un algoritmo di tipo "divide et impera" che dimostra come un ordine n composto da piccoli fattori primi renda l'ECDLP computazionalmente trattabile, indipendentemente dalla grandezza di n stesso.

Il principio matematico dell'attacco è il seguente: dato il problema $Q = kP$, l'algoritmo non cerca di trovare k direttamente. Invece, sfrutta la fattorizzazione di $n = q_1^{e_1} \cdot q_2^{e_2} \cdot \ldots \cdot q_r^{e_r}$ per determinare il valore di k modulo ciascun fattore primo $q_i^{e_i}$. Una volta ottenute tutte le congruenze:

$$k \equiv k_1 \pmod{q_1^{e_1}}, \quad k \equiv k_2 \pmod{q_2^{e_2}}, \quad \ldots, \quad k \equiv k_r \pmod{q_r^{e_r}}$$

L'attaccante può ricombinare queste soluzioni parziali per calcolare in modo efficiente il valore univoco di $k \pmod{n}$. Il cuore dell'attacco consiste nel ridurre il problema principale in una serie di sotto-problemi ECDLP in gruppi di ordine molto più piccolo e quindi deboli.

La complessità totale dell'attacco di Pohlig-Hellman è dominata dal passo più difficile, che è la soluzione dell'ECDLP nel sottogruppo corrispondente al più grande fattore primo di n. La complessità è quindi dell'ordine di $\mathcal{O}(\sqrt{q_{max}})$, dove q_{max} è il più grande fattore primo di n. Questo significa che se tutti i fattori primi di n sono piccoli, l'intero problema può essere risolto rapidamente, rendendo il sistema crittografico insicuro.

Contromisure e la Complessità degli Algoritmi Generici

La vulnerabilità all'attacco di Pohlig-Hellman impone una contromisura non negoziabile: l'ordine n del sottogruppo generato dal punto base B deve essere un numero primo molto grande.

Se n è primo, la sua unica fattorizzazione è n stesso. L'attacco di Pohlig-Hellman non offre alcun vantaggio, poiché il "sotto-problema" da risolvere ha la stessa dimensione del problema originale. Un

crittanalista è quindi costretto a ricorrere ad algoritmi generici, che non sfruttano alcuna proprietà algebrica dell'ordine del gruppo, ma trattano il gruppo come una "scatola nera". I più noti sono:

- **L'Algoritmo Baby-step Giant-step (BSGS):** Questo è un algoritmo deterministico basato su una tecnica "meet-in-the-middle". Per risolvere $Q = kP$, si riscrive $k = i \cdot m + j$ (con $m = \lceil \sqrt{n} \rceil$), riarrangiando l'equazione in $Q - i(mP) = jP$. L'algoritmo pre-calcola e memorizza tutti i possibili valori del lato destro detto baby steps in una struttuara dati consona come una tabella hash. Successivamente, calcola iterativamente i valori del lato sinistro ovvero il giant steps, cercando una corrispondenza nella tabella. La sua complessità temporale e spaziale è di $\mathcal{O}(\sqrt{n})$. L'elevato requisito di memoria lo rende però impraticabile per i parametri crittografici moderni.
- **L'Algoritmo Rho di Pollard:** Questo algoritmo probabilistico risolve l'ECDLP con la stessa complessità temporale di BSGS, $\mathcal{O}(\sqrt{n})$, ma con un requisito di memoria trascurabile ($\mathcal{O}(1)$), rendendolo l'attacco generico di elezione. Si basa sulla ricerca di una collisione in una sequenza pseudo-casuale di punti del gruppo, sfruttando un principio analogo al paradosso del compleanno. In tal senso si cerca di trovare una collisione $P_i = P_j$ che permette di derivare un'equazione lineare in k e risolverla.

4.3 Implicazioni sulla Sicurezza e Confronto con RSA

La complessità di $\mathcal{O}(\sqrt{n})$ degli algoritmi generici è **esponenziale** rispetto alla dimensione in bit (m) della chiave. Poiché $n \approx 2^m$, la complessità è $\mathcal{O}(\sqrt{2^m}) = \mathcal{O}(2^{m/2})$. È questa natura esponenziale che rende l'ECDLP un problema difficile e l'ECC un sistema sicuro.

Al contrario, i migliori algoritmi noti per i problemi alla base di RSA ovvero la fattorizzazione e del DLP classico, come il **General Number Field Sieve (GNFS)**, hanno una complessità **sub-esponenziale**.Questa differenza fondamentale nella difficoltà dei problemi sottostanti implica che per ottenere lo stesso livello di sicurezza, l'ECC richiede chiavi di dimensioni molto più piccole.

Ad esempio si stima che una chiave ECC di **256 bit** offra una resistenza paragonabile a una chiave RSA di **3072 bit**. Questo si traduce in un notevole risparmio di memoria, larghezza di banda e potenza di calcolo, rendendo l'ECC la scelta preferita per ambienti con risorse limitate.

Certamente. Ho compreso la richiesta di migliorare il Capitolo 5, focalizzandomi sull'aumento del rigore espositivo e sull'integrazione esplicita del concetto di "funzione a senso unico con trapola" (one-way trapdoor function), che è il principio cardine della crittografia a chiave pubblica.

Procederò a una riscrittura e amplificazione del capitolo, in particolare espandendo in modo significativo la sezione sull'ECDSA, che nel testo originale era solo accennata. Manterrò la coerenza con lo stile e le

regole del documento.

Assolutamente. La critica è corretta: il capitolo, sebbene completo, presenta ridondanze e una struttura migliorabile. Procedo a una revisione mirata a risolvere queste criticità, accorpando le sezioni semanticamente affini, eliminando le ripetizioni e migliorando il flusso logico del testo, mantenendo inalterato il rigore tecnico.

Capitolo 5: Applicazioni Crittografiche Pratiche delle Curve Ellittiche

La struttura di gruppo abeliano dei punti di una curva ellittica, unita alla difficoltà computazionale dell'ECDLP, costituisce un ambiente ideale per la costruzione di primitive crittografiche a chiave pubblica. L'ECDLP stesso è un esempio paradigmatico di **funzione a senso unico (one-way function)**: dato uno scalare k e un punto P, il calcolo di $Q = kP$ è efficiente; l'operazione inversa, ovvero il calcolo di k dati P e Q, è computazionalmente intrattabile.

Tuttavia, la cifratura asimmetrica e le firme digitali richiedono un meccanismo più sofisticato: una **funzione a senso unico con trappola (trapdoor one-way function)**. Si tratta di una funzione facile da calcolare in una direzione e difficile da invertire, a meno che non si possieda un'informazione segreta — la "trappola" (trapdoor) — che rende l'inversione computazionalmente banale. Nei sistemi basati su ECC, la chiave privata funge da trappola, mentre la chiave pubblica definisce la funzione a senso unico. Questo capitolo analizza i protocolli fondamentali che sfruttano questi principi.

5.1 Scambio di Chiavi Diffie-Hellman su Curve Ellittiche (ECDH)

Il protocollo ECDH è un meccanismo per la negoziazione di chiavi segrete costruito direttamente sulla difficoltà della funzione a senso unico (l'ECDLP) e sulle proprietà algebriche del gruppo. Il suo scopo è permettere a due parti di stabilire una chiave simmetrica condivisa comunicando esclusivamente su un canale insicuro.

5.1.1 Descrizione del Protocollo

1. **Parametri Comuni:** Alice e Bob si accordano pubblicamente sui parametri di dominio della curva (p, a, b, B, n, h).

2. **Generazione Chiavi:** Ciascuna parte genera una coppia di chiavi indipendente. Alice sceglie una chiave privata $d_A \in \{1, \ldots, n-1\}$ e calcola la sua chiave pubblica $Q_A = d_A B$. Analogamente, Bob sceglie d_B e calcola $Q_B = d_B B$.
3. **Scambio:** Alice e Bob si scambiano le rispettive chiavi pubbliche, Q_A e Q_B.
4. **Calcolo del Segreto:** Ciascuna parte combina la propria chiave privata con la chiave pubblica ricevuta:
   - Alice calcola: $S = d_A Q_B = d_A(d_B B) = (d_A d_B)B$
   - Bob calcola: $S = d_B Q_A = d_B(d_A B) = (d_B d_A)B$

Grazie all'associatività e commutatività della moltiplicazione scalare, entrambi ottengono lo stesso punto S. Una Funzione di Derivazione di Chiave (KDF), come una funzione di hash, viene applicata a una rappresentazione canonica di S, come per esempio la sua coordinata, x per generare la chiave simmetrica finale.

Analisi della Sicurezza

Un avversario passivo (Eve) che osserva il canale intercetta i parametri di dominio e le chiavi pubbliche Q_A e Q_B. Per derivare il segreto S, Eve deve risolvere il Problema Computazionale di Diffie-Hellman su Curve Ellittiche (ECDHP) ovvero, dati B, $d_A B$ e $d_B B$, calcolare $(d_A d_B)B$.
Poiché non sono noti algoritmi efficienti per risolvere l'ECDHP senza prima risolvere l'ECDLP che risulta essere intrattabile, il protocollo è sicuro contro l'intercettazione passiva. Tuttavia, ECDH non fornisce autenticazione intrinseca ed è vulnerabile ad attacchi attivi di tipo Man-in-the-Middle (MITM).

5.2 Cifratura Asimmetrica basata su ECC

Lo schema di cifratura ECC-ElGamal è una diretta implementazione del concetto di funzione con trappola. Permette a chiunque conosca la chiave pubblica del destinatario di cifrare un messaggio, ma solo al possessore della chiave privata (la trappola) di decifrarlo.

Incorporazione del Messaggio o Embedding

Una precondizione per la cifratura è la capacità di mappare in modo reversibile un messaggio m a un punto P_m sulla curva. Una mappatura ingenua come $x = m$ fallirebbe in circa il 50% dei casi, poiché non tutti i valori di x corrispondono a un punto dodo che il polinomio $x^3 + ax + b$ deve essere un residuo quadratico modulo p.

Si adotta quindi un approccio probabilistico robusto:

1. **Parametri di Embedding:** Si fissa un piccolo intero di ridondanza, k dove k assume valori simili a $k = 30$.

2. **Generazione Candidati:** Dato m, si genera una sequenza di candidati per la coordinata x: $x_j = mk + j$ per $j = 0, \ldots, k-1$.
3. **Ricerca e Mappatura:** Si itera sui candidati x_j, calcolando $z_j = x_j^3 + ax_j + b \pmod p$. La prima z_j che risulta essere un residuo quadratico, verificato tramite il Criterio di Eulero, $z_j^{(p-1)/2} \equiv 1 \pmod p$, viene utilizzata. Si calcola la sua radice quadrata modulare $y_j = \sqrt{z_j} \pmod p$ per ottenere il punto $P_m = (x_j, y_j)$. La probabilità di fallimento è trascurabile, circa 2^{-k}.
4. **Estrazione o Disembedding:** Il processo inverso è deterministico. Dato un punto $P_m = (x, y)$, si recupera il messaggio con $m = \lfloor x/k \rfloor$.

Architettura del Protocollo

- **Generazione Chiavi (Destinatario, Bob):** Bob sceglie una chiave privata d_B e pubblica la corrispondente chiave pubblica $Q_B = d_B B$.
- **Cifratura (Mittente, Alice):**
  i. Incorpora il messaggio m nel punto P_m.
  ii. Sceglie un intero effimero (nonce) segreto e casuale $r \in \{1, \ldots, n-1\}$, **unico per ogni cifratura**.
  iii. Calcola il testo cifrato $\mathcal{C} = (C_1, C_2)$ dove:

$$C_1 = rB$$

$$C_2 = P_m + rQ_B$$

  iv. Invia $\mathcal{C}$ a Bob.
- **Decifratura (Destinatario, Bob):**
  i. Utilizzando la sua chiave privata d_B (la **trappola**), calcola il punto di mascheramento: $S = d_B C_1 = d_B(rB) = r(d_B B) = rQ_B$.
  ii. Sottrae S da C_2 per annullare il mascheramento e recuperare il punto del messaggio: $P'_m = C_2 - S = (P_m + rQ_B) - rQ_B = P_m$.
  iii. Estrae il messaggio in chiaro da P'_m.

Analisi della Sicurezza

La sicurezza semantica dello schema si fonda sull'intrattabilità dell'ECDHP. Un avversario, in possesso delle informazioni pubbliche (B, Q_B) e del testo cifrato (C_1, C_2), per recuperare P_m dovrebbe essere in grado di calcolare il punto di mascheramento rQ_B. Il calcolo di rQ_B a partire da B, $C_1 = rB$ e $Q_B = d_B B$ è l'ECDHP. La chiave privata d_B agisce come trappola perché fornisce l'unico collegamento computazionalmente efficiente noto tra C_1 e il punto di mascheramento S.

5.3 Algoritmo di Firma Digitale su Curve Ellittiche (ECDSA)

L'ECDSA implementa la firma digitale utilizzando il paradigma della funzione con trappola: solo il possessore della chiave privata (la trappola) può generare una firma valida, ma chiunque con la chiave pubblica può verificarla, garantendo autenticità e non ripudio.

Generazione delle Chiavi

Il firmatario sceglie una chiave privata $d \in \{1, \dots, n-1\}$ e pubblica la corrispondente chiave pubblica $Q = dB$.

Algoritmo di Generazione della Firma

Per firmare un messaggio M:

1. **Hashing:** Calcola $e = \text{HASH}(M)$ e ne deriva un intero z (tipicamente troncando o convertendo e).
2. **Selezione Nonce:** Sceglie un intero effimero (nonce) segreto e casuale $k \in \{1, \dots, n-1\}$. È di importanza critica che k sia unico per ogni firma e mantenuto segreto. Il suo riutilizzo consente il recupero della chiave privata.
3. **Calcolo Componente r:** Calcola il punto $(x_1, y_1) = kB$ e pone $r = x_1 \pmod n$. Se $r = 0$, si sceglie un nuovo k.
4. **Calcolo Componente s:** Calcola $s = k^{-1}(z + rd) \pmod n$. Se $s = 0$, si sceglie un nuovo k.
5. **Firma:** La firma è la coppia (r, s). La chiave privata d è la trappola indispensabile per legare l'hash z al nonce k nell'equazione per s.

Algoritmo di Verifica della Firma

Per verificare una firma (r, s) su un messaggio M usando la chiave pubblica Q:

1. **Verifica Preliminare:** Controlla che $r, s \in [1, n-1]$.
2. **Hashing:** Calcola z dal messaggio M con la stessa funzione usata per la firma.
3. **Calcolo Inverso:** Calcola $w = s^{-1} \pmod n$.
4. **Calcolo Scalari di Verifica:** Calcola $u_1 = zw \pmod n$ e $u_2 = rw \pmod n$.
5. **Calcolo Punto di Verifica:** Calcola il punto $(x_0, y_0) = u_1 B + u_2 Q$. Se il punto è O, la firma non è valida.
6. **Confronto Finale:** La firma è valida se e solo se $x_0 \equiv r \pmod n$.

La coerenza matematica risiede nel fatto che il punto di verifica calcolato è:

$$u_1 B + u_2 Q = (zw)B + (rw)(dB) = (z + rd)wB$$

Sostituendo $w = s^{-1} = k(z + rd)^{-1} \pmod{n}$, si ottiene:

$$(z + rd)(k(z + rd)^{-1})B = kB$$

Il punto calcolato è quindi kB, la cui coordinata x ridotta modulo n è, per definizione, r. Questo conferma la validità della firma.

Capitolo 6: Parametri di Sicurezza e Implementazione

6.1 Selezione dei Parametri della Curva

La sicurezza di un sistema ECC, come abbiamo analizzato dei precedenti capitoli, non dipende solo dalla dimensione della chiave, ma anche dalla scelta oculata dei parametri di dominio della curva. Una scelta errata può rendere il sistema vulnerabile ad attacchi specifici.

I criteri di selezione dei parametri della curva includono:

- **Campo Finito:** Dimensione minima di 256 bit per p o m.
- **Ordine della Curva:** Deve avere un fattore primo n molto grande.
- **Punto Base:** Deve generare un sottogruppo di ordine primo n.
- **Cofattore:** Rapporto $h = |E_p(a, b)|/n$ preferibilmente piccolo ed idealmente $h = 1$.

Per garantire interoperabilità e fiducia, sono state definite diverse **curve standardizzate** da organizzazioni come il NIST (National Institute of Standards and Technology). Esempi noti includono `NIST P-256`, `P-384`, `P-521`, e `Curve25519`, che sono state attentamente selezionate per le loro proprietà di sicurezza.

6.2 Confronto sulla Robustezza Crittografica

Il principale vantaggio pratico dell'ECC risiede nella sua efficienza, che si manifesta nella lunghezza delle chiavi necessarie per raggiungere un determinato livello di sicurezza.

Livello di Sicurezza (bit)	Lunghezza Chiave Simmetrica (bit)	Lunghezza Chiave ECC (bit)	Lunghezza Chiave RSA/DH (bit)
80	80	160	1024
112	112	224	2048
128	128	256	3072
192	192	384	7680
256	256	521	15360

[IMMAGINE: Grafico comparativo che mostra la crescita esponenziale della lunghezza delle chiavi RSA rispetto alla crescita lineare di quelle ECC per livelli di sicurezza crescenti.]

Come si può osservare, la dimensione delle chiavi RSA cresce molto più rapidamente di quella delle chiavi ECC. Questo rende l'ECC particolarmente adatta per ambienti con risorse limitate, come dispositivi mobili, smart card e sistemi IoT.

6.3 Aspetti Implementativi

L'implementazione sicura ed efficiente dell'ECC richiede attenzione a diversi dettagli.

- **Efficienza:** Gli algoritmi di moltiplicazione scalare devono essere implementati in modo ottimale. L'uso di sistemi di coordinate proiettive o Jacobiane può accelerare i calcoli eliminando le costose operazioni di inversione modulare ad ogni passo dell'algoritmo Double-and-Add.
- **Resistenza agli Attacchi a Canale Laterale (Side-Channel Attacks):** Questi attacchi sfruttano le informazioni trapelate durante l'esecuzione fisica di un algoritmo (es. tempo di esecuzione, consumo di energia). Ad esempio, un'implementazione naïf dell'algoritmo Double-and-Add potrebbe rivelare i bit della chiave privata a seconda che venga eseguita o meno l'operazione di "add". Per mitigare questi attacchi, si utilizzano algoritmi a tempo costante, come la "scala di Montgomery", che eseguono sempre la stessa sequenza di operazioni indipendentemente dai bit della chiave.

Conclusioni

La crittografia basata sulle curve ellittiche rappresenta un pilastro della sicurezza informatica moderna. La sua eleganza matematica, fondata sulla struttura di gruppo dei punti di una curva, fornisce le basi per protocolli crittografici robusti ed efficienti. La difficoltà computazionale del Problema del Logaritmo

Discreto su Curve Ellittiche (ECDLP) garantisce un elevato livello di sicurezza con chiavi significativamente più corte rispetto ai sistemi tradizionali come RSA. Questo si traduce in vantaggi tangibili in termini di velocità, consumo di banda e requisiti di memoria, rendendo l'ECC la tecnologia di elezione per un'ampia gamma di applicazioni, specialmente in ambienti con risorse vincolate.

Dall'analisi tecnica dettagliata è emerso come ogni aspetto, dalla scelta del campo e dei parametri della curva fino all'implementazione degli algoritmi, sia cruciale per garantire la sicurezza del sistema. Protocolli come ECDH e schemi di cifratura basati su ElGamal-ECC dimostrano la versatilità e la potenza di questa tecnologia.

Guardando al futuro, la principale minaccia per l'ECC, così come per RSA e altri sistemi a chiave pubblica classici, proviene dall'avvento dei **computer quantistici**. L'**algoritmo di Shor** sarebbe in grado di risolvere l'ECDLP (e la fattorizzazione) in tempo polinomiale, rendendo questi sistemi insicuri. La comunità crittografica è già attivamente impegnata nello sviluppo e nella standardizzazione di una nuova generazione di algoritmi, noti come **crittografia post-quantistica (PQC)**, progettati per resistere agli attacchi sia dei computer classici che di quelli quantistici. Tuttavia, fino a quando i computer quantistici su larga scala non diventeranno una realtà pratica, la crittografia su curve ellittiche continuerà a essere uno degli strumenti più efficaci e affidabili per proteggere le nostre informazioni digitali.

Bibliografia

- Koblitz, N. (1987). *Elliptic Curve Cryptosystems*. Mathematics of Computation, 48(177), 203-209.
- Miller, V. S. (1986). *Use of Elliptic Curves in Cryptography*. In Advances in Cryptology — CRYPTO ’85 Proceedings (pp. 417-426). Springer Berlin Heidelberg.
- Hankerson, D., Menezes, A., & Vanstone, S. (2004). *Guide to Elliptic Curve Cryptography*. Springer-Verlag.
- Silverman, J. H. (2009). *The Arithmetic of Elliptic Curves*. Springer.
- National Institute of Standards and Technology (NIST). (2013). *FIPS PUB 186-4: Digital Signature Standard (DSS)*.