Assolutamente. In qualità di esperto accademico nel campo della crittografia e della matematica applicata, procederò alla redazione del documento di ricerca richiesto, espandendo in modo rigoroso e dettagliato gli appunti forniti. Il documento si atterrà scrupolosamente alla struttura delineata, privilegiando la profondità tecnica e l'analisi matematica rispetto al contesto storico.

Le Curve Ellittiche in Crittografia: Un'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. L'enfasi sarà posta sul rigore matematico, con derivazioni dettagliate delle formule e un'analisi approfondita dei concetti, tralasciando deliberatamente la contestualizzazione storica per concentrarsi esclusivamente sugli aspetti tecnico-scientifici.

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.

Definizione Formale di 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.
   - **Chiusura:** Per ogni $a, b \in K$, la somma $a + b$ appartiene a K.
   - **Associatività:** Per ogni $a, b, c \in K$, vale $(a + b) + c = a + (b + c)$.
   - **Commutatività:** Per ogni $a, b \in K$, vale $a + b = b + a$.
   - **Esistenza dell'Elemento Neutro Additivo:** Esiste un elemento unico $0 \in K$ tale che per ogni $a \in K$, $a + 0 = a$.
   - **Esistenza dell'Inverso Additivo:** Per ogni $a \in K$, esiste un elemento unico $-a \in K$ tale che $a + (-a) = 0$.
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.
   - **Chiusura:** Per ogni $a, b \in K \setminus \{0\}$, il prodotto $a \cdot b$ appartiene a $K \setminus \{0\}$.
   - **Associatività:** Per ogni $a, b, c \in K$, vale $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
   - **Commutatività:** Per ogni $a, b \in K$, vale $a \cdot b = b \cdot a$.
   - **Esistenza dell'Elemento Neutro Moltiplicativo:** Esiste un elemento unico $1 \in K$ (con $1 \neq 0$) tale che per ogni $a \in K$, $a \cdot 1 = a$.
   - **Esistenza dell'Inverso Moltiplicativo:** Per ogni $a \in K \setminus \{0\}$, esiste un elemento unico $a^{-1} \in K$ tale che $a \cdot a^{-1} = 1$.
3. **Proprietà Distributiva:** La moltiplicazione è distributiva rispetto all'addizione.
   - Per ogni $a, b, c \in K$, vale $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$.

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, poiché l'unico assioma non soddisfatto è l'esistenza dell'inverso moltiplicativo per ogni elemento non nullo. 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:

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

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.

1.2 Definizione e Geometria delle Curve Ellittiche

Una curva ellittica non è un'ellisse, ma una curva cubica piana non singolare. Il suo nome 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 (o corta) 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)

Il punto all'infinito O non ha coordinate (x, y) nel piano affine. Può essere visualizzato geometricamente come il punto situato "in cima" e "in fondo" all'asse y, dove tutte le rette verticali si intersecano. Algebricamente, la sua inclusione è necessaria per completare la struttura di gruppo della curva, agendo come elemento neutro per l'operazione di somma. In coordinate proiettive $(X : Y : Z)$, il punto all'infinito corrisponde a $(0 : 1 : 0)$.

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 è alla base della legge di gruppo, è la seguente:
Una retta non verticale interseca una curva ellittica in al massimo tre punti.
Sostituendo l'equazione di una retta $y = \lambda x + \nu$ nell'equazione della curva, si ottiene:

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

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

Questa è un'equazione cubica in x, che ha al massimo tre soluzioni nel campo K (contando le molteplicità). Una retta verticale $x = c$ interseca la curva in due punti $(c, \sqrt{c^3 + ac + b})$ e $(c, -\sqrt{c^3 + ac + b})$, e si considera che intersechi anche il punto all'infinito O, per un totale di tre intersezioni. Questa proprietà garantisce che, dati due punti, la retta che li unisce ne identificherà quasi sempre un terzo, che è il mattone fondamentale per definire l'operazione di somma.

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 (Somma) 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 (sommando P per $k - 1$ volte) sarebbe inefficiente per valori grandi di k. Si utilizza invece un algoritmo molto più rapido.

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 un computer è approssimata tramite numeri in virgola mobile. Le operazioni crittografiche richiedono una precisione assoluta; qualsiasi errore di arrotondamento distruggerebbe l'integrità dei calcoli.
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

computazionalmente intrattabili, che sono il fondamento della sicurezza crittografica.

3.2 Curve Ellittiche su $\mathbb{Z}_p$ (Campi Primi)

Una curva ellittica su un campo finito primo $\mathbb{Z}_p$ (con $p > 3$ primo) è definita in modo analogo al caso reale, ma tutte le operazioni sono eseguite 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 $\gcd(d, p) = 1$. 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:

$$s \cdot d + t \cdot p = \gcd(d, p)$$

Se $\gcd(d, p) = 1$, 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:

- $23 = 3 \cdot 7 + 2$
- $7 = 3 \cdot 2 + 1$
- $2 = 2 \cdot 1 + 0 \implies \gcd(23, 7) = 1.$

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 di Addizione in $\mathbb{Z}_p$

Consideriamo la curva $y^2 \equiv x^3 - x + 1 \pmod{67}$ e i punti $P = (17, 41)$ e $Q = (27, 48)$.
Verifichiamo prima che appartengano alla curva:

- Per P: $41^2 = 1681 \equiv 6 \pmod{67}$. $17^3 - 17 + 1 = 4913 - 16 = 4897 \equiv 6 \pmod{67}$. Il punto P è sulla curva.
- Per Q: $48^2 = 2304 \equiv 26 \pmod{67}$. $27^3 - 27 + 1 = 19683 - 26 = 19657 \equiv 26 \pmod{67}$. Il punto Q è sulla curva.

Calcoliamo $S = P + Q$. Usiamo le formule di addizione modulo 67.

$$\lambda \equiv (y_Q - y_P)(x_Q - x_P)^{-1} \pmod{67}$$

$$\lambda \equiv (48-41)(27-17)^{-1} \equiv 7 \cdot 10^{-1} \pmod{67}$$

Dobbiamo calcolare $10^{-1} \pmod{67}$. Con l'algoritmo di Euclide Esteso: $67 = 6 \cdot 10 + 7, 10 = 1 \cdot 7 + 3, 7 = 2 \cdot 3 + 1$. Lavorando a ritroso si ottiene $1 = (-2) \cdot 67 + (13) \cdot 10$. Quindi $10^{-1} \equiv 13 \pmod{67}$.

$$\lambda \equiv 7 \cdot 13 = 91 \equiv 24 \pmod{67}$$

Ora calcoliamo le coordinate di $S = (x_S, y_S)$:

$$x_S \equiv \lambda^2 - x_P - x_Q \equiv 24^2 - 17 - 27 \equiv 576 - 44 = 532 \equiv 59 \pmod{67}$$

$$y_S \equiv \lambda(x_P - x_S) - y_P \equiv 24(17-59) - 41 \equiv 24(-42) - 41 \pmod{67}$$

$$y_S \equiv 24(25) - 41 = 600 - 41 = 559 \equiv 23 \pmod{67}$$

Attenzione: $y_S = -y_R$. La formula corretta è $y_S = \lambda(x_P - x_S) - y_P$.

$$y_S \equiv 24(17-59) - 41 \equiv 24(-42) - 41 \equiv 24(25) - 41 = 600 - 41 = 559 \pmod{67}$$

$559 = 8 \cdot 67 + 23$, quindi $y_S \equiv 23 \pmod{67}$.
Ma la somma è il riflesso, quindi $P + Q = (59, -y_R)$. L'ordinata del punto somma è:

$$y_S \equiv \lambda(x_P - x_S) - y_P \equiv 24(17-59) - 41 \equiv 24(-42) - 41 \equiv 24(25) - 41 \pmod{67}$$

$$y_S \equiv 600 - 41 = 559 \pmod{67} \equiv 23 \pmod{67}$$

.
Il punto somma è $(59, -23 \pmod{67}) = (59, 44 \pmod{67})$.
Rifacendo il calcolo con le formule corrette:
$x_S = 59$.
$y_S = -(y_P + \lambda(x_S - x_P)) = -(41 + 24(59-17)) = -(41 + 24(42)) = -(41 + 1008)$.
$1008 \equiv 3 \pmod{67}$.
$y_S \equiv -(41 + 3) = -44 \equiv 23 \pmod{67}$.
Quindi $P + Q = (59, 23)$. L'esempio negli appunti aveva un errore di calcolo.

Ordine di una Curva e Ordine di un Punto

L'**ordine** di una curva ellittica $E_p(a, b)$, denotato $|E_p(a, b)|$, è il numero totale di punti che essa contiene, incluso O. Questo valore è cruciale per la sicurezza. Il **Teorema di Hasse** fornisce un intervallo per l'ordine della curva:

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

L'**ordine di un punto** P è il più piccolo intero positivo n tale che $nP = O$. Per il Teorema di Lagrange, l'ordine di qualsiasi punto deve essere un divisore dell'ordine della curva. In crittografia, si cercano curve il cui ordine sia un numero primo grande, o abbia un fattore primo molto grande. Si sceglie poi un punto generatore B il cui ordine n sia questo grande numero primo. Il gruppo utilizzato per la crittografia non è l'intera curva $E_p(a, b)$, ma il sottogruppo ciclico generato da B, che ha n elementi.

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

Oltre ai campi primi, un'altra famiglia di campi finiti molto usata in crittografia è quella dei **campi binari** (o di 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.

4.1 Definizione e Natura dell'ECDLP

Il Problema del Logaritmo Discreto (DLP) "classico" è definito nel gruppo moltiplicativo di un campo finito $\mathbb{Z}_p^*$. Dati un generatore g e un elemento h, il problema consiste nel trovare l'intero k tale che $h \equiv g^k \pmod p$.

L'**ECDLP** è l'analogo 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 P sulla curva di ordine n, e un altro punto Q che è un multiplo scalare di P (cioè $Q \in \langle P \rangle$), 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**.

L'ECDLP è un esempio perfetto di **funzione one-way**:

- **Direzione Facile:** Dati k e P, calcolare $Q = kP$ è computazionalmente facile. Si può fare efficientemente usando l'algoritmo Double-and-Add in tempo $\mathcal{O}(\log k)$.
- **Direzione Difficile:** Dati P e Q, trovare k è computazionalmente intrattabile per curve e parametri scelti opportunamente.

4.2 Difficoltà Computazionale dell'ECDLP

La ragione per cui l'ECC è così attraente è che i migliori algoritmi noti per risolvere l'ECDLP sono significativamente meno efficienti dei migliori algoritmi per risolvere i problemi su cui si basano RSA (fattorizzazione) e Diffie-Hellman classico (DLP).

Gli algoritmi più noti per risolvere l'ECDLP in un gruppo generico (cioè senza sfruttare particolari debolezze della curva) sono:

- **Baby-step Giant-step:** Un algoritmo deterministico con complessità temporale e spaziale di $\mathcal{O}(\sqrt{n})$, dove n è l'ordine del punto base.
- **Pollard's Rho Algorithm:** Un algoritmo probabilistico con complessità temporale $\mathcal{O}(\sqrt{n})$ ma con requisiti di memoria trascurabili.

Esistono altri algoritmi come Pohlig-Hellman che sono efficaci solo se l'ordine n del gruppo ha fattori primi piccoli. Per questo motivo, in crittografia si sceglie sempre un punto base B il cui ordine n sia un numero primo molto grande.

La complessità di questi algoritmi è **esponenziale** nella dimensione in bit dell'ordine del gruppo (se l'ordine n ha circa m bit, $\sqrt{n} \approx 2^{m/2}$). Al contrario, i migliori algoritmi per la fattorizzazione (come il General Number Field Sieve) hanno una complessità **sub-esponenziale**. Questa differenza fondamentale implica che per ottenere lo stesso livello di sicurezza, l'ECC richiede chiavi di dimensioni molto più piccole rispetto a RSA.

Ad esempio, una chiave ECC da 256 bit offre un livello di sicurezza paragonabile a una chiave RSA da 3072 bit. Questo si traduce in un notevole risparmio di memoria, larghezza di banda e potenza di calcolo.

Capitolo 5: Applicazioni Crittografiche Pratiche delle Curve Ellittiche

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

Il protocollo di scambio di chiavi Diffie-Hellman può essere adattato per utilizzare il gruppo dei punti di una curva ellittica. Il risultato, noto come ECDH, è uno dei metodi più diffusi per stabilire una chiave segreta condivisa su un canale insicuro.

Protocollo ECDH

1. **Accordo sui Parametri Pubblici:** Alice e Bob si accordano pubblicamente su un insieme di parametri, chiamati "parametri di dominio della curva ellittica":
   - Una curva ellittica $E_p(a, b)$ su un campo finito $\mathbb{Z}_p$.
   - Un punto base (o generatore) B sulla curva.
   - L'ordine n del punto B, che deve essere un numero primo grande.
2. **Generazione delle Chiavi:**
   - **Alice** sceglie un intero segreto casuale n_A (la sua **chiave privata**), con $1 \leq n_A < n$. Calcola poi il punto $P_A = n_A B$ (la sua **chiave pubblica**).
   - **Bob** sceglie un intero segreto casuale n_B (la sua **chiave privata**), con $1 \leq n_B < n$. Calcola poi il punto $P_B = n_B B$ (la sua **chiave pubblica**).
3. **Scambio e Calcolo della Chiave Condivisa:**
   - Alice invia la sua chiave pubblica P_A a Bob.
   - Bob invia la sua chiave pubblica P_B ad Alice.
   - **Alice**, usando la sua chiave privata n_A e la chiave pubblica di Bob P_B, calcola il punto segreto S:

$$S = n_A P_B = n_A(n_B B) = (n_A n_B)B$$

- **Bob**, usando la sua chiave privata n_B e la chiave pubblica di Alice P_A, calcola il punto segreto S:

$$S = n_B P_A = n_B(n_A B) = (n_B n_A)B$$

Grazie all'associatività della moltiplicazione scalare, Alice e Bob ottengono lo stesso punto S senza mai scambiarsi le loro chiavi private. La coordinata x del punto S (o una sua derivata tramite una funzione di hash) viene tipicamente usata come chiave simmetrica per successive comunicazioni cifrate.

Sicurezza del Protocollo

Un attaccante (Eve) che osserva la comunicazione può intercettare i parametri pubblici ($E_p(a, b), B, n$) e le chiavi pubbliche scambiate (P_A, P_B). Per calcolare la chiave segreta S, Eve dovrebbe risolvere l'ECDLP:

- Dati B e $P_A = n_A B$, Eve dovrebbe trovare n_A.
- Oppure, dati B e $P_B = n_B B$, Eve dovrebbe trovare n_B.
  Poiché risolvere l'ECDLP è computazionalmente intrattabile, il protocollo è sicuro contro attacchi passivi. Tuttavia, come il protocollo Diffie-Hellman classico, l'ECDH è vulnerabile ad attacchi attivi **Man-in-the-Middle (MITM)** se le chiavi pubbliche non sono autenticate (ad esempio, tramite certificati digitali).

5.2 Crittografia a Chiave Pubblica con Curve Ellittiche (Schema di Cifratura ElGamal-ECC)

È possibile costruire un sistema di cifratura a chiave pubblica analogo al sistema ElGamal, utilizzando le curve ellittiche.

5.2.1 Trasformazione del Messaggio in Punto

Il primo passo per cifrare un messaggio m è mapparlo in modo reversibile a un punto P_m sulla curva ellittica. Questa non è un'operazione banale, poiché non tutti i valori di x corrispondono a un punto sulla curva (il valore $x^3 + ax + b$ deve essere un residuo quadratico modulo p).

L'**Algoritmo di Koblitz** (semplificato) è un approccio probabilistico per risolvere questo problema.

1. **Parametri:** Si fissa un piccolo intero h (es. $h = 20$ o $h = 30$). Il messaggio m deve essere un intero tale che $(m + 1)h < p$.
2. **Mappatura:** Per un dato messaggio m, si costruiscono h candidati per l'ascissa x:

$$x_i = mh + i, \quad \text{per } i = 0, 1, \ldots, h - 1$$

3. **Ricerca:** Per ogni x_i, si calcola $z_i = x_i^3 + ax_i + b \pmod p$. Si verifica se z_i è un residuo quadratico modulo p. Questo può essere fatto efficientemente calcolando il simbolo di Legendre: $z_i^{(p-1)/2} \equiv 1 \pmod p$.
4. **Estrazione Radice:** Se si trova un x_i per cui z_i è un residuo quadratico, si calcola la sua radice quadrata $y_i = \sqrt{z_i} \pmod p$. Esistono algoritmi efficienti per questo calcolo (es. Algoritmo di Tonelli-Shanks).
5. **Punto Messaggio:** Il punto P_m è (x_i, y_i). La probabilità di successo è alta, poiché circa la metà dei numeri in $\mathbb{Z}_p$ sono residui quadratici. Se $h = 30$, la probabilità di fallimento è circa $1/2^{30}$.
6. **Decodifica:** Per recuperare il messaggio dal punto $P_m = (x, y)$, il destinatario calcola semplicemente:

$$m = \lfloor \frac{x}{h} \rfloor$$

5.2.2 Protocollo di Cifratura e Decifratura

1. **Accordo Pubblico:** Mittente e destinatario si accordano sui parametri di dominio $(E_p(a, b), B, n)$ e sul parametro h per la codifica.
2. **Chiavi del Destinatario (Bob):** Bob genera una chiave privata n_B e una chiave pubblica $P_B = n_B B$.
3. **Fase di Cifratura (Alice):**
   - Alice trasforma il suo messaggio m nel punto P_m usando l'algoritmo di Koblitz.
   - Sceglie un intero casuale segreto r (diverso per ogni messaggio).
   - Calcola due punti che costituiscono il testo cifrato:
     - $V = rB$
     - $W = P_m + rP_B$
   - Alice invia la coppia di punti $\langle V, W \rangle$ a Bob.
4. **Fase di Decifratura (Bob):**
   - Bob riceve la coppia $\langle V, W \rangle$.
   - Usando la sua chiave privata n_B, calcola:

$$n_B V = n_B(rB) = r(n_B B) = rP_B$$

   - Sottrae questo punto da W per recuperare P_m:

$$W - n_B V = (P_m + rP_B) - (rP_B) = P_m$$

   - Una volta ottenuto $P_m = (x, y)$, Bob recupera il messaggio originale calcolando $m = \lfloor x/h \rfloor$.

La sicurezza di questo schema si basa sull'intrattabilità del Problema Computazionale di Diffie-Hellman su Curve Ellittiche (ECDHP), che è strettamente correlato all'ECDLP.

5.3 Cenni alle Firme Digitali su Curve Ellittiche (ECDSA)

L'**Elliptic Curve Digital Signature Algorithm (ECDSA)** è l'adattamento dell'algoritmo di firma digitale DSA al contesto delle curve ellittiche. È ampiamente utilizzato in molteplici applicazioni, tra cui le transazioni Bitcoin e la comunicazione sicura TLS.
Il processo di firma coinvolge la chiave privata del firmatario e un numero casuale per generare una coppia di valori (r, s) che costituiscono la firma. La verifica, invece, utilizza la chiave pubblica del firmatario e i parametri pubblici della curva per confermare l'autenticità e l'integrità del messaggio. Il vantaggio principale di ECDSA rispetto a RSA o DSA è la dimensione molto ridotta delle firme, che a parità di sicurezza sono significativamente più corte, con benefici in termini di storage e larghezza di banda.

Capitolo 6: Parametri di Sicurezza e Implementazione

6.1 Selezione dei Parametri della Curva

La sicurezza di un sistema ECC 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 fondamentali includono:

1. **Dimensione del Campo:** Il numero primo p (per campi primi) o l'esponente m (per campi binari) devono essere sufficientemente grandi da rendere gli attacchi come Pollard's Rho impraticabili. Tipicamente si usano dimensioni di almeno 256 bit.
2. **Ordine della Curva:** L'ordine della curva, $|E_p(a, b)|$, non deve essere primo, ma deve avere un fattore primo n molto grande.
3. **Punto Base e suo Ordine:** Il punto base B deve generare un sottogruppo di ordine n, dove n è un numero primo grande. Questo previene l'attacco di Pohlig-Hellman. Il **cofattore** $h = |E_p(a, b)|/n$ dovrebbe essere piccolo (idealmente $h = 1$).
4. **Evitare Curve Speciali:** Alcune famiglie di curve, come le curve **supersingolari**, sono vulnerabili all'attacco MOV (Menezes-Okamoto-Vanstone), che riduce l'ECDLP a un DLP in un campo di estensione, dove può essere risolto più facilmente. Pertanto, tali curve devono essere evitate.

Per garantire interoperabilità e fiducia, sono state definite diverse **curve standardizzate** da organizzazioni come il NIST (National Institute of Standards and Technology) e Brainpool. 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 (Internet of Things).

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)*.