Elliptic Curve Cryptography

CHENG-HAN TSAI

Spring 2025

Contents

1 Introduction

1.1 DLP

Discrete Logarithm Problem (DLP), which plays a role in cryptography, is a mathematical problem. The **DLP** can be defined as follows:

Given a finite cyclic multiplicative group $\mathbb{Z}_p$ where p is a prime number, a generator $g \in \mathbb{Z}_p^*$, and an element $h \in \mathbb{Z}_p^*$. Find an integer x such that

$$g^x \equiv h \ (mod\ p)$$

If p is large enough, no algorithm available today can efficiently compute this x. Therefore, **DLP** becomes a basis of cryptography and is widely used in the Diffie-Hellman (DH) Key Exchange, ElGamal Encryption, or Digital Signature Algorithm (DSA).

1.2 ECC and ECDLP

To achieve faster performance, shorter key lengths, and stronger security strength, **Elliptic Curve Cryptography (ECC)** was developed. At the core of **ECC** lies the **Elliptic Curve Discrete Logarithm Problem (ECDLP)**, which is the variant of elliptic curve of the classical DLP. The **ECDLP** is defined as follows:

Given a group of points on an elliptic curve E. Finding an integer k such that $Q = kP$, where P and Q are two points on the curve E.

Since **ECC** offers higher security strength and shorter key lengths, it is well-suited for modern cybersecurity requirements. Moreover, no efficient algorithm is currently known that can solve the **ECDLP** in polynomial time, which ensures the security of **ECC**.

2 Mathematics Background

2.1 Elliptic Curves over $\mathbb{R}$

Definition 2.1. *Let $a, b \in \mathbb{R}$ such that $\Delta = 4a^3 + 27b^2 \neq 0$. An **elliptic curve over** $\mathbb{R}$ is the set of points (x, y) with $x, y \in \mathbb{R}$ which satisfy the equation*

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

*together with a single element denoted $\mathcal{O}$ and called the **point at infinity**. That is, the **elliptic curve over** $\mathbb{R}$ is the set*

$$E(\mathbb{R}) = \{(x, y) \in \mathbb{R} \times \mathbb{R} : y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}.$$

Remark 2.1. *The discriminant is not zero ($\Delta = 4a^3 + 27b^2 \neq 0$) means that the graph has no **cusps**, **self-intersections**, or **isolated points**.*

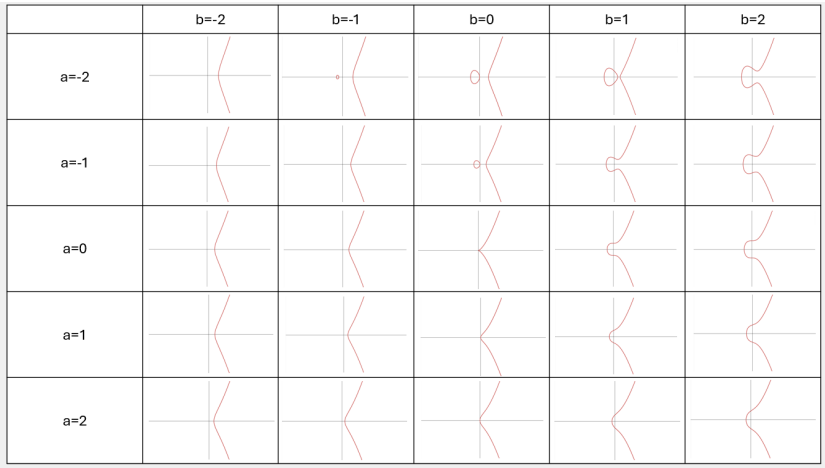

Figure 1: Examples for $y^2 = x^3 + ax + b$

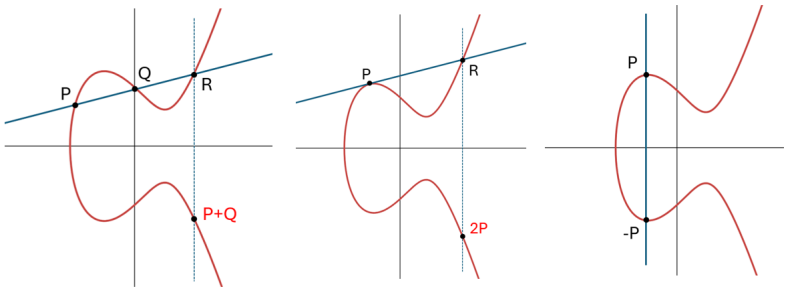

Figure 2: Geometric illustration of the binary operation on $E(\mathbb{R})$

Example 2.1. *As illustrated in Figure 1.*

In order to define a group structure over $E(\mathbb{R})$, it is necessary to introduce a binary operation that satisfies the group axioms. Therefore, we will consider the geometry of elliptic curves, and determine a suitable "addition".

Example 2.2. *As illustrated in Figure 2, a binary operation $+$ on the set of points $E(\mathbb{R})$ must satisfy the closure property in order to define a group structure. To ensure this, given two points $P, Q \in E(\mathbb{R})$, we define their sum $P + Q$ as follows:*

- *If $P \neq Q$, consider the* ***secant line*** *passing through P and Q;*
- *If $P = Q$, consider the* ***tangent line*** *to the curve at P.*

In both cases, the corresponding line intersects the elliptic curve at a third point R. Since elliptic curves are defined by a cubic equation, ***Bézout's the-***

orem *guarantees that such a line intersects the curve in exactly three points (counting multiplicities) in the projective plane. However, we do* ***not*** *define the sum as $P+Q=R$. Instead, we reflect the point R over the x-axis to obtain the point $-R$, and define the result of the addition as:*

$$P+Q:=-R.$$

This definition aligns with the symmetry of the elliptic curve: if (x,y) lies on the curve, so does $(x,-y)$. The reflection ensures the result remains on the curve, and it provides a consistent geometric interpretation for addition.

We now define the binary operation algebraically:

Definition 2.2. *Let $E(\mathbb{R})$ be an elliptic curve defined over the field $\mathbb{R}$. For any points $P=(x_1,y_1)$ and $Q=(x_2,y_2)$ in $E(\mathbb{R})$, the* ***point addition operation*** *$+$ is defined as follows:*

1. ***(Identity)*** *$P+\mathcal{O}=P$ and $\mathcal{O}+P=P$.*
2. ***(Inverse)*** *If $x_1=x_2$ and $y_1=-y_2$, then $P+Q=\mathcal{O}$.*
3. ***(Addition of distinct points)*** *If $P\neq Q$ and $x_1\neq x_2$, define*

$$\lambda=\frac{y_2-y_1}{x_2-x_1},$$

then compute

$$x_3=\lambda^2-x_1-x_2,\quad y_3=\lambda(x_3-x_1)+y_1,$$

and set

$$P+Q=(x_3,-y_3).$$

4. ***(Point doubling)*** *If $P=Q$ and $y_1\neq 0$, define*

$$\lambda=\frac{3x_1^2+a}{2y_1},$$

then compute

$$x_3=\lambda^2-2x_1,\quad y_3=\lambda(x_3-x_1)+y_1,$$

and set

$$2P=P+P=(x_3,-y_3).$$

Example 2.3. *Consider the elliptic curve*

$$E:y^2=x^3-5x+8$$

and the point $P=(1,2)$ lies on the curve E. Using the tangent line construction, we compute

$$2P=\left(-\frac{7}{4},-\frac{27}{8}\right).$$

Using the secant line construction, we find

$$3P = P + 2P = \left(\frac{553}{121}, -\frac{11950}{1331}\right).$$

Similarly,

$$4P = \left(\frac{45313}{11664}, -\frac{8655103}{1259712}\right).$$

Observe that the coordinates increase significantly with repeated additions.

Having established a complete addition law on the points of $E(\mathbb{R})$, we now construct the group structure as follows:

Proposition 2.1. *The set of points $E(\mathbb{R})$, together with the point addition operation $+$, forms a group, which is called the* ***elliptic curve group****.*

Proof. To verify that $(E(\mathbb{R}), +)$ forms a group, we check the group axioms:

1. **Closure:** Given any points $P, Q \in E(\mathbb{R})$, the sum $P + Q$ is defined as the reflection across the x-axis of the third point of intersection between the elliptic curve and the line passing through P and Q. Therefore, $P + Q \in E(\mathbb{R})$.
2. **Associativity:** The proof is omitted.
3. **Identity:** The point at infinity $\mathcal{O}$ acts as the identity element under addition.
4. **Inverses:** For every point $P = (x, y) \in E(\mathbb{R})$, the point $-P = (x, -y) \in E(\mathbb{R})$ satisfies $P + (-P) = \mathcal{O}$.

Hence, $(E(\mathbb{R}), +)$ forms a group. □

Proposition 2.2. *The elliptic curve group is an abelian group.*

Proof. Let $P, Q \in E(\mathbb{R})$. Consider the line L passing through P and Q. The line L intersects E at exactly one more point R. By definition,

$$P + Q = -R,$$

where $-R$ denotes the reflection of R about the x-axis.

Since the line through P and Q is the same as the line through Q and P, the third intersection point R is identical for both $P + Q$ and $Q + P$. Consequently,

$$P + Q = Q + P,$$

which proves that the addition operation is commutative.

Hence, the group $(E(\mathbb{R}), +)$ is abelian. □

2.2 Elliptic Curves over $\mathbb{F}_p$

If an elliptic curve is defined over the real numbers $\mathbb{R}$, then the set of points $E(\mathbb{R})$ contains infinitely many elements. However, such a representation is not practical for implementation on computers, as digital systems can only handle finite, discrete structures. Therefore, we will focus on elliptic curves used in cryptography—specifically, **elliptic curves defined over the finite field** $\mathbb{F}_p$, which consists of p elements.

Definition 2.3. *Let $a, b \in \mathbb{R}$ such that $4a^3 + 27b^2 \pmod{p} \neq 0$. An **elliptic curve over** $\mathbb{F}_p$ is the set*

$$E(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p \mid y^2 \equiv x^3 + ax + b \pmod{p}\} \cup \{\mathcal{O}\}.$$

Example 2.4. *Consider the elliptic curve over the finite field $\mathbb{F}_{37}$:*

$$E : y^2 = x^3 - 5x + 8 \pmod{37}.$$

Let the points $P = (6, 3)$ and $Q = (9, 10)$ lie on $E(\mathbb{F}_{37})$. Using the point addition operation, we compute the following values in $E(\mathbb{F}_{37})$:

$$\begin{aligned} 2P &= (35, 11), \\ 3P &= (34, 25), \\ 4P &= (8, 6), \\ 5P &= (16, 19), \\ P + Q &= (11, 10), \\ 3P + 4Q &= (31, 28), \quad \textit{and so on.} \end{aligned}$$

All coordinates are computed modulo 37, and each resulting point remains within the group $E(\mathbb{F}_{37})$.

Example 2.5. *By substituting each possible value $x = 0, 1, 2, \ldots, 36$ and checking whether the expression*

$$x^3 - 5x + 8 \mod 37$$

is a quadratic residue modulo 37, we find the following points on the elliptic curve:

$$E : y^2 = x^3 - 5x + 8 \pmod{37}.$$

The set $E(\mathbb{F}_{37})$ consists of the following 45 points (including the point at infinity $\mathcal{O}$):

$$\begin{aligned} &(1, \pm 2),\ (5, \pm 21),\ (6, \pm 3),\ (8, \pm 6),\ (9, \pm 27),\ (10, \pm 25), \\ &(11, \pm 27),\ (12, \pm 23),\ (16, \pm 19),\ (17, \pm 27),\ (19, \pm 1),\ (20, \pm 8), \\ &(21, \pm 5),\ (22, \pm 1),\ (26, \pm 8),\ (28, \pm 8),\ (30, \pm 25),\ (31, \pm 9), \\ &(33, \pm 1),\ (34, \pm 25),\ (35, \pm 26),\ (36, \pm 7),\ \mathcal{O}. \end{aligned}$$

There are 9 points whose order divides 3. Therefore, the group of points can be expressed, as an abstract group, as:

$$E(\mathbb{F}_{37}) \cong \mathbb{Z}_3 \times \mathbb{Z}_{15}.$$

From the explicit enumeration of points on the curve $E(\mathbb{F}_{37})$, we observe that the group contains a subgroup of order 3, as well as elements generating a subgroup of order 15. This illustrates a fundamental structural property of elliptic curve groups over finite fields, which is formally stated in the following theorem:

Theorem 2.1. *The set of points $E(\mathbb{F}_p)$ forms a finite abelian group, which is either a cyclic group or the product of two cyclic groups.*

Proof. (sketch)
1. By *Prop.2.2* and the fact that $\mathbb{F}_p$ is finite, $E(\mathbb{F}_p)$ is an finite abelian group.
2. By the **fundamental theorem of finitely generated abelian groups**, any finite abelian group can be expressed as a finite direct product of cyclic groups, and this decomposition is unique.

$$\mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2} \oplus \cdots \oplus \mathbb{Z}_{n_r}, \quad \text{where } n_i \mid n_{i+1}$$

3. In each $\mathbb{Z}_{n_i}$, there are exactly n_1 elements whose order divides n_1. Hence, the total number of elements in the entire group whose order divides n_1 is at most n_1^r. However, we can prove the number of points of order dividing n_1 is at most n_1^2. Therefore, we must have the inequality $n_1^r \leq n_1^2$, which implies $r \leq 2$. Thus, the number of generators of the group $E(\mathbb{F}_p)$ is at most 2.

□

Again, from the explicit enumeration of points on $E(\mathbb{F}_{37})$, we observe that although the points are discrete, their overall distribution is not random. The following theorem provides a precise upper and lower bound on the number of points on $E(\mathbb{F}_{37})$:

Lemma 2.1. *Let A be an abelian group, and let*

$$d : A \to \mathbb{Z}$$

be a positive definite quadratic form. Then for all $\psi, \phi \in A$,

$$|d(\psi - \phi) - d(\phi) - d(\psi)| \leq 2\sqrt{d(\phi)d(\psi)}.$$

Proof. For $\psi, \phi \in A$, define

$$L(\psi, \phi) = d(\psi - \phi) - d(\phi) - d(\psi)$$

to be the bilinear form associated with the quadratic form d. Since d is positive definite, for all integers $m, n \in \mathbb{Z}$, we have

$$0 \leq d(m\psi - n\phi) = m^2 d(\psi) + mnL(\psi, \phi) + n^2 d(\phi).$$

In particular, taking

$$m = -L(\psi, \phi) \quad \text{and} \quad n = 2d(\psi)$$

yields

$$0 \leq d(\psi)\left(4d(\psi)d(\phi) - L(\psi,\phi)^2\right).$$

This gives the desired inequality, provided that $\psi \neq 0$, while for $\psi = 0$ the inequality is trivial. □

Theorem 2.2 (Hasse's Theorem)**.** *Let $E : y^2 = x^3 + ax + b$ be an elliptic curve defined over the finite field $\mathbb{F}_p$ with $a, b \in \mathbb{F}_p$. Then*

$$\left| \#E(\mathbb{F}_p) - (p+1) \right| \leq 2\sqrt{p}.$$

Proof. Choose a Weierstrass equation for E with coefficients in $\mathbb{F}_p$, and let

$$\varphi : E \to E, \quad (x, y) \mapsto (x^p, y^p),$$

be the p-th power Frobenius morphism. Since the Galois group $G_{\overline{\mathbb{F}}_p/\mathbb{F}_p}$ is (topologically) generated by the p-th power map on $\overline{\mathbb{F}}_p$, we see that for any point $P \in E(\overline{\mathbb{F}}_p)$,

$$P \in E(\mathbb{F}_p) \text{ if and only if } \varphi(P) = P.$$

Thus,

$$E(\mathbb{F}_p) = \ker(1 - \varphi),$$

and find that

$$\#E(\mathbb{F}_q) = \#\ker(1-\varphi) = \deg(1-\varphi).$$

Since the degree map on $\mathrm{End}(E)$ is a positive definite quadratic and since $\deg \varphi = q$, *lemma.2.1* gives the desired result. □

2.3 ECDLP

Recall that the **Elliptic Curve Discrete Logarithm Problem (ECDLP)** is defined as follows:

Given a group of points on an elliptic curve E. Finding an integer k such that $Q = kP$, where P and Q are two points on the curve E.

k is called the **discrete logarithm (or index) of Q with respect to P**, denoted by:

$$m = \log_P(Q) = \mathrm{ind}_P(Q).$$

Let n be the order of P in the group $E(\mathbb{F}_p)$. Then

$$\log_P : \langle P \rangle \to \mathbb{Z}_n$$

is a group isomorphism, which is the inverse of the map

$$k \mapsto kS.$$

Fundamentally, the difficulty of the **ECDLP** arises from the computational hardness of determining the discrete logarithm. In other words, given points P and Q on an elliptic curve, finding the discrete logarithm $k = \log_P(Q)$ such that

$$Q = kP$$

has no known polynomial-time algorithm. This problem is similar to the classical **DLP** in number theory, but the elliptic curve structure makes it significantly more challenging.

3 Solutions of ECDLP

3.1 Exhaustive Search Method

Consider the **Exhaustive Search Method** to solve the **ECDLP**. We compute multiples of the point P on the elliptic curve over $\mathbb{F}_p$:

$$k_1P \quad k_2P, \quad k_3P, \quad \dots$$

for randomly chosen integers $k_1, k_2, k_3, \dots$ until we find an integer k such that

$$kP = Q.$$

Let n be the order of the point P. In the worst case, this method requires n iterations. On average, it will find the solution after approximately $n/2$ iterations. Therefore, the time complexity is $O(n)$. This method does not require additional storage and has constant space complexity $O(1)$. However, for cryptographically relevant parameters, where p is a large prime (e.g., $p \approx 2^{256}$), exhaustive search is computationally infeasible.

Example 3.1. *Consider the elliptic curve*

$$E : y^2 = x^3 + 2x + 2$$

defined over the finite field $\mathbb{F}_{17}$. *Let* $P = (5, 1) \in E(\mathbb{F}_{17})$. *We wish to find an integer* k *such that*

$$Q = (6, 3) = kP.$$

We proceed by computing successive multiples of P*:*

$$1P = (5, 1)$$
$$2P = (6, 3)$$

We see that $2P = Q$*, so the solution is* $k = 2$.

3.2 Collision Search Method

Consider the **Collision Search Method**, which is a probabilistic algorithm based on the birthday paradox. The algorithm works as follows:

- Choose random integers $k_1, k_2, k_3, \dots$ and compute:

  - List 1: $k_1P, k_2P, k_3P, \ldots$
  - List 2: $Q - k_1P, Q - k_2P, T - Q_3P, \ldots$
- Continue until a collision is found: $k_iP = Q - k_jP$.
- Then we have $k_i + k_j = k$, which solves $Q = kP$.

By the birthday paradox, the expected number of iterations before a collision occurs (time complexity) is $O(\sqrt{p})$, since the number of points on the elliptic curve satisfies $|E(\mathbb{F}_p)| = O(p)$. This method is significantly faster than **Exhaustive Search Method** (which takes $O(p)$ time), but still becomes impractical for cryptographically large prime fields. It forms the conceptual basis of more efficient algorithms like **Baby-Step Giant-Step** and **Pollard's Rho**.

Example 3.2. *Consider the elliptic curve defined over the finite field $\mathbb{F}_{17}$:*

$$E : y^2 = x^3 + 2x + 2 \pmod{17}$$

*Given the point $P = (5, 1)$ and the target point $Q = (6, 3)$, we want to find an integer k such that $Q = kP$. The procedure of **Collision Search Method** is:*

- *Compute multiples of P:*

$$1P = (5, 1), \quad 2P = (6, 3), \quad \ldots$$

- *Construct two lists:*
  - *List 1: k_iP for randomly chosen integers k_i.*
  - *List 2: $Q - k_jP$ for randomly chosen integers k_j.*
- *From the calculations, $2P = (6, 3) = Q$, thus $k = 2$.*

Therefore, the discrete logarithm of Q with respect to P is 2.

Remark 3.1 (Birthday Paradox)**.** *Assume we randomly select k elements from a set of size N, and we want to estimate the probability of at least one collision (i.e., two elements being the same). The probability that all k elements are distinct is:*

$$P(\textit{no collision}) = \frac{N}{N} \cdot \frac{N-1}{N} \cdot \frac{N-2}{N} \cdots \frac{N-k+1}{N} = \prod_{i=0}^{k-1} \left(1 - \frac{i}{N}\right)$$

When $k \ll N$, this product can be approximated using the exponential function:

$$P(\textit{no collision}) \approx e^{-k^2/(2N)}$$

Therefore, the probability of at least one collision is:

$$P(\textit{collision}) \approx 1 - e^{-k^2/(2N)}$$

To find the value of k that results in a collision probability of approximately 0.5, we solve:

$$1 - e^{-k^2/(2N)} = 0.5$$

This yields:

$$e^{-k^2/(2N)} = 0.5 \quad \Rightarrow \quad \frac{k^2}{2N} = \ln 2 \quad \Rightarrow \quad k \approx \sqrt{2N \ln 2} \approx 1.177\sqrt{N}$$

Hence, it takes approximately $1.2\sqrt{N}$ random samples to have a 50% chance of finding a collision. This phenomenon underpins the efficiency of collision-based algorithms like the Birthday Attack in cryptography.

3.3 Baby-Step Giant-Step Algorithm

The **Baby-Step Giant-Step (BSGS)** algorithm is a deterministic method for solving the **ECDLP**. Let $E(\mathbb{F}_p)$ be an elliptic curve defined over the finite field $\mathbb{F}_p$, and let $P \in E(\mathbb{F}_p)$ be a point of order n. Given another point $Q \in \langle P \rangle$, the goal is to find an integer $k \in \mathbb{Z}_n$ such that:

$$Q = kP.$$

The algorithm proceeds as follows:

1. Let $k = \lceil \sqrt{n} \rceil$.
2. **Baby steps:** Compute and store the list
$$L_1 = \{jP \mid j = 0, 1, \ldots, k-1\}.$$
3. **Giant steps:** For $i = 0, 1, \ldots, k-1$, compute
$$Q - i \cdot (kP),$$
and check whether it matches any entry in L_1.
4. When a collision is found, i.e., $jP = Q - i(kP)$, we solve for the discrete logarithm as
$$\log_P(Q) = ik + j.$$

The time complexity of the Baby-Step Giant-Step (BSGS) algorithm is $O(\sqrt{n})$, which is significantly faster than the exhaustive search with time complexity $O(n)$. However, the baby-step phase requires storing the list

$$L_1 = \{jP \mid j = 0, 1, \ldots, k-1\},$$

resulting in a space complexity of $O(\sqrt{n})$. This algorithm is practical for moderate values of n, but becomes memory-intensive for large key sizes.

Example 3.3. *Consider the elliptic curve defined over the finite field* $\mathbb{F}_{17}$*:*

$$E : y^2 = x^3 + 2x + 2,$$

with the base point$P = (5, 1)$ *and the point* $Q = (6, 3)$*. The goal is to find the integer* k *such that* $Q = kP$*. The procedure of* ***BSGS algorithm*** *is:*

1. *Suppose the order of* P *is* $n = 19$*. Compute* $m = \lceil \sqrt{n} \rceil = 5$*.*
2. ***Baby steps:*** *Compute and store the list* $L_1 = \{jP \mid j = 0, 1, 2, 3, 4\}$*.*
3. ***Giant steps:*** *For* $i = 0, 1, \ldots, 4$*, compute* $Q - i(mP)$ *and check for collisions with elements in* L_1*.*
4. *When a collision is found, i.e., for some* i, j*,* $jP = Q - i(mP)$*, solve for* k *as* $k = im + j$*.*

For example, if a collision occurs at $i = 2$ *and* $j = 3$*, then* $k = 2 \times 5 + 3 = 13$*, which satisfies* $Q = kP$*.*

3.4 Pollard's ρ Method

Pollard's ρ method is a probabilistic algorithm designed to solve the **ECDLP**. Given an elliptic curve $E(\mathbb{F}_p)$, a point $P \in E(\mathbb{F}_p)$ of order n, and another point $Q = kP$, the goal is to determine the integer k. This method constructs a pseudo-random sequence of points on the curve of the form:

$$X_i = a_i P + b_i Q$$

where $a_i, b_i \in \mathbb{Z}_n$.

Using **Floyd's cycle detection algorithm** (also known as the **tortoise and hare method**), the algorithm searches for a collision $X_i = X_j$ for $i \neq j$. When a collision is found, we have:

$$\begin{aligned} & a_i P + b_i Q = a_j P + b_j Q \\ \Rightarrow \quad & (a_i - a_j)P = (b_j - b_i)Q \\ \Rightarrow \quad & (a_i - a_j)P = (b_j - b_i)kP \\ & \Rightarrow \quad k \equiv \frac{a_i - a_j}{b_j - b_i} \pmod{n} \end{aligned}$$

provided that $b_j \not\equiv b_i \pmod{n}$ and that $b_j - b_i$ has a multiplicative inverse modulo n.

The time complexity of Pollard's ρ method is $O(\sqrt{n})$, based on the birthday paradox, where n is the order of P. Since the algorithm only requires tracking a small number of variables for the current and previous points in the sequence, its space complexity is $O(1)$.

This method is significantly more space-efficient than the Baby-Step Giant-Step algorithm, which requires $\mathcal{O}(\sqrt{n})$ space. Pollard's ρ method is well-suited for cryptographic applications with limited memory but may require more time in practice due to its probabilistic nature.

Example 3.4. *Let $E(\mathbb{F}_{19})$ be an elliptic curve over the finite field $\mathbb{F}_{17}$, and let $P \in E$ be a point of order $n = 19$. Suppose we are given a point $Q = kP$, and our goal is to find the integer $k \in \mathbb{Z}_{19}$. We define the initial values:*

$$X_0 = a_0P + b_0Q = 0P + 1Q = Q, \quad a_0 = 0, \quad b_0 = 1$$

Divide the group into three subsets based on $x \bmod 3$, and define the update rules:

- *S_1: If $x \bmod 3 = 0$, then $X_{i+1} = X_i + P, \quad a_{i+1} = a_i + 1, \quad b_{i+1} = b_i$.*
- *S_2: If $x \bmod 3 = 1$, then $X_{i+1} = 2X_i, \quad a_{i+1} = 2a_i, \quad b_{i+1} = 2b_i$.*
- *S_3: If $x \bmod 3 = 2$, then $X_{i+1} = X_i + Q, \quad a_{i+1} = a_i, \quad b_{i+1} = b_i + 1$.*

Use Floyd's cycle detection (tortoise and hare) to detect a collision $X_i = X_j, \quad i \neq j$. Assume a collision is found:

$$X_i = 13P + 6Q, \quad X_j = 4P + 2Q$$

Then

$$\begin{aligned} 13P + 6Q &= 4P + 2Q \\ (13 - 4)P &= (2 - 6)Q \\ 9P &= (-4)Q \end{aligned}$$

$$9P = -4kP \Rightarrow 9 = -4k \mod 19$$

Since modular inverse is $4^{-1} \mod 19 = 5$, we have

$$k \equiv -9 \cdot 5 \mod 19 = -45 \mod 19 = 11$$

Remark 3.2 (Floyd's cycle detection algorithm)**.** ***Floyd's cycle detection algorithm*** *is a classic technique used to detect cycles in sequences generated by iterative functions, and it is crucial in Pollard's ρ method for detecting collisions efficiently without storing all previous elements.*

Given a function f that generates a sequence:

$$x_0, x_1 = f(x_0), x_2 = f(x_1), \ldots,$$

the goal is to find if the sequence contains a cycle, i.e., find indices μ and $\lambda > 0$ such that:

$$x_\mu = x_{\mu+\lambda} = x_{\mu+2\lambda} = \cdots.$$

The algorithm uses two pointers:

- ***Tortoise****: moves one step at a time: $x_{i+1} = f(x_i)$.*
- ***Hare****: moves two steps at a time: $x_{i+1} = f(f(x_i))$.*

If a cycle exists, the hare will eventually meet the tortoise inside the cycle, i.e.,

$$x_i = x_{2i}.$$

Floyd's algorithm detects this collision in $O(\mu + \lambda)$ time with only $O(1)$ space.

4 Conclusions and Extensions

4.1 Conclusions

The table below compares four classical algorithms for solving the **Elliptic Curve Discrete Logarithm Problem (ECDLP)**: Exhaustive Search, Collision Search, Baby-Step Giant-Step (BSGS), and Pollard's ρ Method:

Algorithm	Type	Time Complexity	Space Complexity
Exhaustive Search	Deterministic	$O(n)$	$O(1)$
Collision Search	Probabilistic	$O(\sqrt{p})$	$O(\sqrt{p})$
BSGS	Deterministic	$O(\sqrt{n})$	$O(\sqrt{n})$
Pollard's ρ	Probabilistic	$O(\sqrt{n})$	$O(1)$

Table 1: Comparison of ECDLP Solving Algorithms

Notice that n denotes the order of the point $P \in E(\mathbb{F}_p)$, that is, the size of the subgroup $\langle P \rangle$ generated by P, while p represents the size of the finite field $\mathbb{F}_p$ over which the elliptic curve is defined. Below is a summary of the main characteristics of each method:

- **Exhaustive Search Method**: Mainly used for small-scale or didactic purposes due to its poor efficiency.
- **Collision Search Method**: Historically important but largely obsolete compared to BSGS and Pollard's ρ.
- **Baby-Step Giant-Step Algorithm**: Fast but memory-intensive and best suited for moderate-size problems.
- **Pollard's ρ Method**: Often preferred in practice for its low memory usage, despite being probabilistic.

4.2 Extensions: Finite Field on Computer

4.2.1 Introduction

Finite field arithmetic is foundational in cryptography, coding theory, and many algebraic computations. Specifically, operations over finite fields such as $\mathbb{F}_p$ and $\mathbb{F}_{2^m}$ are core to algorithms like RSA, ECC, and AES. However, implementing such operations efficiently on computers is non-trivial due to the required modular arithmetic and non-native representations. This section analyzes the challenges of performing finite field computations on a digital computer.

4.2.2 Challenges of Arithmetic in $\mathbb{F}_p$

In a finite field $\mathbb{F}_p$, where p is a prime, all arithmetic operations must be done modulo p. For example:

$$a + b \mod p, \quad a - b \mod p, \quad a \cdot b \mod p$$

Modulo Computation requires extra computation, especially when p is large

Moreover, division in $\mathbb{F}_p$ is defined as multiplication by the multiplicative inverse. For $a \in \mathbb{F}_p$, the inverse a^{-1} satisfies:

$$a \cdot a^{-1} \equiv 1 \mod p$$

Computing the inverse typically uses the **Extended Euclidean Algorithm (EEA)**, which is significantly more complex than addition or multiplication.

4.2.3 Challenges in $\mathbb{F}_{2^m}$

Fields of the form $\mathbb{F}_{2^m}$ are widely used in symmetric cryptography and ECC over binary fields. Elements are represented as bit-strings or binary polynomials.

- **Addition:** Bitwise XOR is easy.
- **Multiplication:** Polynomial multiplication modulo an irreducible polynomial is much more computationally intensive and complex to implement than addition, especially when m is large.
- **Inversion:** Uses **EEA** or **Fermat's theorem**, which is still computationally expensive.

Arithmetic in $\mathbb{F}_{2^m}$ is harder to implement efficiently on standard CPUs and often requires custom hardware or optimized software libraries.

4.3 Extensions: Shor's Algorithm

Shor's algorithm is a quantum algorithm that efficiently solves the discrete logarithm problem and integer factorization in polynomial time. This is also a solution for ECDLP. The algorithm exploits **Quantum Parallelism** and the **Quantum Fourier Transform** to find the period of a certain function related to the group operation on the elliptic curve. This period corresponds directly to the discrete logarithm k.

Classical algorithms for the discrete logarithm problem require exponential time, while Shor's algorithm runs in polynomial time $O(\log^3 p)$ for prime p.

The security of Elliptic Curve Cryptography (ECC) is fundamentally based on the assumption that classical computers cannot efficiently solve the ECDLP. However, Shor's algorithm presents a significant future threat: once large-scale quantum computers become practical, existing ECC systems will no longer be secure.

Therefore, it is strongly recommended that the cryptographic community actively research and adopt **post-quantum cryptographic** schemes. These schemes are designed to remain secure even in the presence of quantum adversaries, providing a necessary safeguard against emerging quantum-era risks.

5 References

[1] J. H. Silverman, "An Introduction to the Theory of Elliptic Curves", presentation, University of Wyoming Summer School on Computational Number Theory and Applications to Cryptography, Laramie, WY, Jun.–Jul. 2006.

[2] M. Holec, "Babystep-Giantstep Algorithm and Solution of Elliptic Curve Discrete Logarithm Problem", M.Sc. thesis, Dept. of Computer Systems, Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic, Jun. 29, 2018.
[3] S. P. Lokhande, I. Gupta, and D. B. Kulkarni, "A review: Solving ECDLP problem using Pollard's Rho algorithm," International Journal of Advanced Research in Computer and Communication Engineering, vol. 6, no. 5, pp. 424–427, May 2017.
[4] Joseph H. Silverman, "The Arithmetic of Elliptic Curves", 2nd ed. Springer, 2009.