Chapter 7

Elliptic Curve and Pairing

7.1 Introduction

Over the past few decades Elliptic curve have found useful applications. The curve offers rich and insightful structure, especially those defined over a finite field. These curves are suitable for wide variety of applications in cryptography. There is not much work done in the area of secret sharing, where elliptic curve can be effectively utilized. In this chapter we explore the fundamentals of elliptic curve and then an important construct called Bilinear pairing, which can be effectively utilized to build secret sharing schemes with several extended capabilities. Our aim in this section is to summarize just enough of the basic theory of elliptic curve needed for secret sharing applications. The readers may refer to books and survey articles to learn the theory of elliptic curve in detail [23] [88] [119] [121] [147] [195] [196]. Elliptic curve pairing and their applications are reviewed by Dutta et al [67].
Chapter 7. Elliptic Curve and Pairing

7.2 Elliptic Curves

An elliptic curve is the set of solutions to an equation of the form

\[ Y^2 = X^3 + AX + B \]

Equations of this type are called \textit{Weierstrass equations} after the mathematician who studied them extensively. Two elliptic curves \( E_1 \) and \( E_2 \) defined by the equations

\[ E_1 : Y^2 = X^3 - 3X + 3 \quad \text{and} \quad E_2 : Y^2 = X^3 - 6X + 3 \]

The plot of these curves are given in 7.1 and 7.2.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure1.png}
\caption{Elliptic Curve E1}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure2.png}
\caption{Elliptic Curve E2}
\end{figure}
An amazing feature of elliptic curve is that, we can take two points on an elliptic curve and add them to produce a third point in a natural way. The addition operation is visualized geometrically. If we connect two points $P$ and $Q$ on the curve with a line $L$, it will intersect the curve at a third point $R$. The reflection $R'$ of it will be the sum $P + Q$. The Figure 7.3 shows elliptic curve point addition geometrically. If we add the point $P = (a, b)$ to its reflexion about the X-axis $P' = (a, -b)$, the line $L$ through these two points will be a vertical line $x = a$ and this line will intersect the curve at two points. There is no third point of intersection. The solution for this is to consider a point $O$ that does not exist in the XY-plane, but we assume that it lies on every vertical line. So

$$P + P' = O$$

It is also noted that, if we add the point $P$ to $O$, we get back to $P$. i.e.,

$$P + O = P$$

So $O$ acts like zero for elliptic curve addition, which is the identity element in elliptic curve additive group.
Definition 7.2.1. An elliptic curve $E$ is the set of solutions to a Weierstrass equation

$$E : Y^2 = X^3 + AX + B$$

with an extra point $O$. The constant $A$ and $B$ must satisfy

$$4A^3 + 27B^2 \neq 0$$

The addition law on $E$ is defined in the following way. Let $P$ and $Q$ be two points on $E$ and let $L$ be the line connecting $P$ and $Q$. If $P = Q$ then $L$ is a tangent to $E$ at $P$. The intersection of $E$ and $L$ consist of three points $P, Q, R$, counted with appropriate multiplicities and the assumption
that $O$ lies on every vertical line. If $R = (a, b)$, then the sum of $P$ and $Q$ denoted by $P + Q$, is the reflection $R' = (a, -b)$ of $R$ across X-axis.

If $P = (a, b)$, then $-P = (a, -b)$, which is the reflected point. Repeated addition is represented as multiplication of a point by an integer. i.e.;

$$[k]P = \underbrace{P + P + P + \cdots + P}_k \text{ points}$$

Similarly

$$[-k]P = \underbrace{-P - P - P - \cdots - P}_k \text{ points}$$

The quantity $\Delta_E = 4A^3 + 27B^2$ is called the discriminant of $E$. When $\Delta_E \neq 0$, the cubic polynomial $X^3 + AX + B$ does not have any repeated roots or we say that the curve $E$ is smooth. i.e., $E$ can be factored as

$$X^3 + AX + B = (X - a_1)(X - a_2)(X - a_3)$$

where $a_1, a_2, a_3$ are distinct.

**Theorem 7.2.1.** Let $E$ be an elliptic curve. Then the addition law on $E$ has the following properties.

1. $P + O = O + P = P$ for all $P \in E$. (Identity)
2. $P + (-P) = O$ for all $P \in E$. (Inverse)
3. $(P + Q) + R = P + (Q + R)$ for all $P, Q, R \in E$. (Associative)
4. $P + Q = Q + P$ for all $P, Q \in E$. (Commutative)

It is noted that, the addition law makes the points of $E$ into an abelian group. The proof of these laws can be found in [131] [195].

We can find explicit formulas for easy addition and subtraction of points on an elliptic curve.
Theorem 7.2.2. Let $E$ be an elliptic curve

$$E : Y^2 = X^3 + AX + B$$

and suppose we want to add two distinct points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$. Let the line connecting $P_1$ and $P_2$ to be

$$L : Y = \lambda X + C$$

The slope and $y$ intercept of the line is given by

$$\lambda = \begin{cases} 
\frac{y_2 - y_1}{x_2 - x_1}, & \text{if} \ P_1 \neq P_2. \\
\frac{3x_1^2 + A}{2y_1}, & \text{if} \ P_1 = P_2.
\end{cases}$$

and let

$$C = y_1 - \lambda x_1.$$ 

and let

$$x_3 = \lambda^2 - x_1 - x_2, \quad \text{and} \quad y_3 = \lambda(x_1 - x_3) - y_1.$$ 

Then $P_1 + P_2 = (x_3, y_3)$

- if $P_1 = O$, then $P_1 + P_2 = P_2$.
- if $P_2 = O$, then $P_1 + P_2 = P_1$.
- if $x_1 = x_2$ and $y_1 = -y_2$, then $P_1 + P_2 = O$.

Proof. The line $L$ intersect the curve in three points. Let $P_3 = (x_3, y_3)$ be the third zero of $L$. Now substitute $Y = \lambda X + C$ into the equation of $E$ to obtain

$$(\lambda X + C)^2 = X^3 + AX + B$$

Expanding this will give

$$f(X) = X^3 - \lambda^2 X^2 + (A - 2C\lambda)X + B - C^2 = 0$$
7.3. Elliptic Curves Over Finite Fields

\( x_1, x_2, x_3 \) must be the roots of this equation, since they are the \( x \) co-
ordinates of \( P_1, P_2 \) and \( P_3 \), which satisfy both the equation of the elliptic
curve and \( L = 0 \). Thus \( f(X) : (X - x_1)(X - x_2)(X - x_3) \). Comparing the
coefficients of the \( X^2 \) terms gives \( x_1 + x_2 + x_3 = \lambda^2 \). Hence \( x_3 = \lambda^2 - x_1 - x_2 \).
We can write \( y'_3 = \lambda x_3 + C \). So \( y'_3 = \lambda x_1 + y_1 - \lambda x_1 \). By taking the inverse
of \( y'_3 \), i.e., \(-y'_3\), we will obtain \( y_3 = \lambda(x_1 - x_3) - y_1 \).

7.3 Elliptic Curves Over Finite Fields

Elliptic curves whose points have coordinates in a finite field \( \mathbb{F}_p \) are the
best candidates for cryptography. We can define an elliptic curve over \( \mathbb{F}_p \)
by the equation

\[
E : Y^2 = X^3 + AX + B
\]

where \( A, B \in \mathbb{F}_p \) with \( 4A^3 + 27B^2 \neq 0 \) and \( p \geq 3 \). We then look for points
\((x, y) \in \mathbb{F}_p \) satisfying the elliptic curve equation. i.e.,

\[
E(\mathbb{F}_p) = \{(x, y) : x, y \in \mathbb{F}_p \text{ which satisfy } y^2 = x^3 + Ax + B\} \cup \{O\}
\]

Example 7.3.1. Let us consider an elliptic curve

\[
E : Y^2 = X^3 + 3X + 8 \text{ over the field } \mathbb{F}_{13}.
\]

The points on this curve can be found out by putting in all possible
values of \( X \) and then check whether \( Y \) is quadratic residue or not.
Corresponding to each \( X \) value, there will be two possible values for \( Y \). So
there can be a maximum of \( 2p + 1 \) points on the curve including \( O \). In the
example given, \( E(\mathbb{F}_{13}) \) consist of nine points.

\[
E(\mathbb{F}_{13}) = \{O, (1, 5), (1, 8), (2, 3), (2, 10), (9, 6), (9, 7), (12, 2), (12, 11)\}
\]
Chapter 7. Elliptic Curve and Pairing

The theory of geometry developed for the field \( \mathbb{R} \) can be used for \( \mathbb{F}_p \) using the algebraic geometry. If \( P = (x_1, y_1) \) and \( Q = (x_2, y_2) \) are the two points then the sum \( R = P + Q = (x_3, y_3) \), can be obtained by applying the formulas mentioned in addition theorem 7.2.2. The division operation can be done by finding the inverse of the number in the field. All operations are done in \( \mathbb{F}_p \).

**Theorem 7.3.1.** [99] Let \( E \) be an elliptic curve over \( \mathbb{F}_p \) and let \( P \) and \( Q \) be points on \( E(\mathbb{F}_p) \).

(a) The elliptic curve addition algorithm applied to \( P \) and \( Q \) yields a point \( R = P + Q \) in \( E(\mathbb{F}_p) \).

(b) The addition law on \( E(\mathbb{F}_p) \) satisfies all the properties listed in the theorem 7.2.1. It is noted that the addition law makes \( E(\mathbb{F}_p) \) into a finite group.

The congruence \( X^3 + AX + B \equiv 0 \pmod{p} \) is quadratic residue 50% of the time or else it is a non residue or 0 (happens only once). Hasse’s theorem provides a bound on the number of points on an elliptic curve.

**Theorem 7.3.2.** (Hasse) Let \( N \) be the number of points in an elliptic curve (\( \#E(\mathbb{F}_p) \)) defined over \( \mathbb{F}_p \). Then

\[
N - (p + 1) \leq t_p \quad \text{with} \quad t_p \text{ satisfying } |t_p| \leq 2\sqrt{p}
\]

The quantity \( t_p \) is called the trace of Frobenius for \( E(\mathbb{F}_p) \). Hasse’s theorem gives a bound for number of points on the elliptic curve. However it will not provide a method for calculating the number of points. The method of substituting each value for \( X \) and checking \( X^3 + AX + B \) is a square or not against a table take \( O(p) \) time, so is very inefficient. An algorithm developed by Schoof [188] can be used to find \( \#E(\mathbb{F}_p) \) in \( O((\log p)^6) \) time. Elkies and Atkin improved this algorithm later and is
known as **SEA algorithm** [187] [188]. Satoh [185] developed a reasonably efficient algorithm for counting points on curve in the field of form $p^e$, where $p$ is a small prime and $e$ is moderately large.

### 7.4 Elliptic Curve Discrete Logarithm Problem

Elliptic Curve Discrete Logarithm Problem (ECDLP) is computationally harder than the Discrete Logarithm Problem (DLP) in $\mathbb{F}_p^*$. Let $E(\mathbb{F}_p)$ be an elliptic curve defined over $\mathbb{F}_p$. Given two points $Q$ and $P$, the attacker has to find out $n$ such that $Q = nP$. i.e., the attacker has to find out how many times $P$ must be added to itself in order to get $Q$.

$$Q = P + P + P + \cdots + P = nP$$

**Definition 7.4.1.** Let $E$ be an elliptic curve over the finite field $\mathbb{F}_p$ and let $P$ and $Q$ be the points in the $E(\mathbb{F}_p)$. Then the Elliptic Curve Discrete Logarithm Problem (ECDLP) is the problem of finding an integer $n$ such that $Q = nP$. The integer $n$ is represented as

$$n = log_P(Q)$$

We refer $n$, the elliptic curve logarithm of $Q$ with respect to $P$.

There are situations, where $n$ is not defined i.e., $Q$ is not a multiple of $P$. But in the cryptographic applications, we choose $P$ and compute $Q = nP$. So $n = log_P(Q)$ always exist. If $s$ is the order of $P$ such that $sP = O$. The value of $n$ will be in $\mathbb{Z}/s\mathbb{Z}$. The ECDL satisfies

$$log_P(Q_1 + Q_2) = log_P(Q_1) + log_P(Q_2)$$
7.5 Hardness of ECDLP

The collision algorithms take approximately \( \sqrt{N} \) steps in order to find a collision among \( N \) objects. But they require creation of one or more lists of size approximately \( \sqrt{N} \). The baby step-giant step algorithm is a collision algorithm that is used to solve the discrete logarithm problem for the field \( \mathbb{F}_p \) in \( \sqrt{p} \) time. The index calculus method solves the DLP in \( \mathbb{F}_p \) much more rapidly. But for elliptic curve groups, collision algorithm is the fastest known method. There is no index calculus algorithm known for ECDLP and indeed, there are no general algorithms known that solve ECDLP in less than \( O(p) \) steps. This is the reason elliptic curves groups are found useful at present.

In order to solve the ECDLP, the attacker can build two lists of points by randomly choosing integers \( a_1, a_2, \ldots, a_r \) and \( b_1, b_2, \ldots, b_r \) between 1 and \( p \).

\[
\begin{align*}
List - 1 & : a_1P, a_2P, a_3P, \ldots, a_rP \\
List - 2 & : b_1P + Q, b_2P + Q, b_3P + Q, \ldots, b_rP + Q
\end{align*}
\]

As soon as a collision or match is found between two lists then ECDLP can be solved. If some \( a_iP = b_jP + Q \), then \( Q = (a_i - b_j)P \), which provides the solution. If \( r \) is larger than \( \sqrt{p} \) i.e.; \( r \approx 3\sqrt{p} \), then there is a very good chance of collision. The collision algorithm usually requires a lot of storage for the two lists. Pollard’s \( \rho \) method can be used for storage-free collision algorithm with the same running time.

There are some primes \( p \) for which the DLP in \( \mathbb{F}_p^* \) is comparatively easy. For example if \( p - 1 \) is a product of small primes, then the Pohling-Hellman algorithm \([170]\) gives a quick solution to DLP in \( \mathbb{F}_p^* \). In a similar way, there are some elliptic curve and some primes for which ECDLP in \( E(\mathbb{F}_p) \) is comparatively easy. These curves must be avoided while building secure crypto systems.
7.6 Computing $nP$, Double and Add Algorithm

In cryptographic application, we need to compute $nP$. If $n$ is large then computing $nP$ by $P, 2P, 3P, 4P, \ldots, nP$ is not practical. The most efficient way to compute $nP$ is similar to computing $g^e$ using square and multiply algorithm. Since the operation on elliptic curve involves point addition, we call it as square and add. The underlying technique is as follows.

First write $n$ in binary form

$$n = n_0 + n_1.2 + n_2.2^2 + n_3.2^3 + \cdots + n_r.2^r$$

with $n_0, n_1, \ldots, n_r \in \{0, 1\}$

**Algorithm 7.1:** Double and Add Algorithm for Elliptic Curve.

- **Input:** Point $P \in E(\mathbb{F}_p)$ and $n \geq 1$
- **Output:** The new point $R = nP \in E(\mathbb{F}_p)$

1. Set $Q = P$ and $R = O$
2. while $n > 0$ do
  3. if $n \equiv 1 \pmod{2}$ set $R = R + Q$
  4. set $Q = 2Q$ and $\lfloor n = n/2 \rfloor$
5. end
6. Return the point $R$, which equals $nP$.

Next we compute

$$Q_0 = P, \quad Q_1 = 2Q_0, \quad Q_2 = 2Q_1, \ldots, Q_r = 2Q_{r-1}$$

Each $Q_i$ is twice the previous $Q_{i-1}

$$Q_i = 2^i P$$

Computing each $Q_i$ needs a doubling and a total of $r$ doubling. Finally the computation of $nP$ needs additional $r$ additions also.

$$nP = n_0Q_0 + n_1Q_1 + n_2Q_2 + \cdots + n_rQ_r$$
If we consider a point addition as a point operation in $E(\mathbb{F}_p)$. Then computing $nP$ needs $2r$ point operations in $E(\mathbb{F}_p)$. Since $n \geq 2^r$, the computation takes not more than $2\log_2(n)$ point operation to compute $nP$. The double and add algorithm is given in Algorithm 7.1.

### 7.7 Elliptic Curve Over $\mathbb{F}_{p^k}$

The binary is the most suitable language for computers. Using an elliptic curve modulo 2 is the preferred one. But it is noted that $E(\mathbb{F}_2)$ contains at most five points, so $E(\mathbb{F}_2)$ is not suitable for the security applications.

Field containing $2^k$ elements is a preferred choice. It is noted that for every prime power $p^k$, there exist a field $\mathbb{F}_{p^k}$ with $p^k$ elements. We can consider an elliptic curve whose Weierstrass equation has coefficients in a field $\mathbb{F}_{p^k}$, and then consider the points having coordinates in $\mathbb{F}_{p^k}$. Hasse’s theorem is applicable in this more generalized settings also.

#### Theorem 7.7.1. (Hasse)

Let $E$ be an elliptic curve over $\mathbb{F}_{p^k}$. Then

$$\#E(\mathbb{F}_{p^k}) = p^k + 1 - t_{p^k} \text{ with } t_{p^k} \text{ satisfying } |t_{p^k}| \leq 2^{p^k/2}$$

This shows that elliptic curve over $\mathbb{F}_{2^k}$ is a suitable choice for cryptographic application. But the discriminant $\Delta = -16(4A^3 + 27B^2)$ is always zero. The solution is to use a more generalized Weierstrass equations to define the elliptic curve.

#### Definition 7.7.1.

An Elliptic curve $E$ is the set of solutions to a generalized Weierstrass equation

$$E : Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$$

together with the point $O$. The coefficients $a_1, a_2, \ldots, a_6$ should satisfy $\Delta \neq 0$, to ensure that the curve is non singular. $\Delta$ is defined as
7.7. Elliptic Curve Over $\mathbb{F}_{p^k}$

$$\Delta = -b_2^2 b_8 - 8b_4^3 - 27b_6^2 + 9b_2 b_4 b_6$$

where

$$b_2 = a_1^2 + 4a_2$$
$$b_4 = 2a_4 + a_1 a_3$$
$$b_6 = a_3^2 + 4a_6$$
$$b_8 = a_1^2 a_6 + 4a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2$$

The addition law can be applied here also. But the reflection map $(x, y) \rightarrow (x, -y)$ is replaced by

$$(x, y) \rightarrow (x, -y - a_1 x - a_3)$$

If $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are the two points with $P_1 \neq \pm P_2$. Then the sum $P_3 = (x_3, y_3)$, where

$$x_3 = \lambda^2 + a_1 \lambda - a_2 - x_1 - x_2 \quad \text{with} \quad \lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

If $P_1 = P_2$ then the $x$ coordinates of $P_3$ is

$$x_3 = \frac{x^4 - b_4 x^2 - 2b_6 x - b_8}{4x^3 + b_2 x^2 + 4b_4 x + b_6}$$

There are lot of computational advantage when working with elliptic curves defined over $\mathbb{F}_{2^k}$. For security reasons $k$ should be prime. If $k$ is composite then for $j|k$, there exist a subfield $\mathbb{F}_{p^j}$, which can be used to speed up computations by compromising security. The use of an elliptic curve in $\mathbb{F}_2$, with coordinates of the points chosen from $\mathbb{F}_{2^k}$ allows to use Frobenius map instead of doubling map, which provides significant gain in efficiency.
Definition 7.7.2. The Frobenius map $\tau$ is the map from the field $\mathbb{F}_{p^k}$ to itself defined by the rule

$$\tau : \mathbb{F}_{p^k} \to \mathbb{F}_{p^k}, \quad \alpha \to \alpha^p$$

It is noted that the Forbenius map preserves addition and multiplication.

$$\tau(\alpha + \beta) = \tau(\alpha) + \tau(\beta)$$

and

$$\tau(\alpha \cdot \beta) = \tau(\alpha) \cdot \tau(\beta)$$

The multiplication rule is straightforward

$$\tau(\alpha \cdot \beta) = (\alpha \cdot \beta)^p = \alpha^p \cdot \beta^p = \tau(\alpha) \cdot \tau(\beta)$$

For $p = 2$ the addition law is easy

$$\tau(\alpha + \beta) = (\alpha + \beta)^2 = \alpha^2 + 2\alpha \cdot \beta + \beta^2 = \alpha^2 + \beta^2 = \tau(\alpha) + \tau(\beta)$$

Let $P = (x, y) \in E(\mathbb{F}_{2k})$ be a point on $E$, with coordinates in some larger field $\mathbb{F}_{2k}$. The Forbenius map is defined by applying $\tau$ to each coordinate.

$$\tau(P) = (\tau(x), \tau(y)) \in E(\mathbb{F}_{2k})$$

If $P$ and $Q$ are the elements of $E(\mathbb{F}_{2k})$, then

$$\tau(P + Q) = \tau(P) + \tau(Q)$$

This shows that Forbenius map is a group homomorphism of $E(\mathbb{F}_{2k})$ to itself. The computation of $nP$ mentioned in section 7.6 needs approximately $\log n$ doubling and $\frac{1}{2} \log n$ additions. A refinement which uses negative powers of 2 reduces the time to approximately $\log n$
doubling and $\frac{1}{3} \log n$ additions. In both the cases, the number of doubling remains $\log n$. Kolbitz suggested an idea to replace doubling map with the Forbenius map. This lead to large saving in time because the computation of $\tau(P)$ takes comparatively less time than computing $2P$.

7.8 Points of Finite Order on Elliptic Curves

The points of finite order on an elliptic curve are called torsion points. They play a major role in forming the elliptic curve groups.

**Definition 7.8.1.** Let $m \geq 1$ be an integer. A point $P \in E$ satisfying $mP = O$ is called point of order $m$ in the group $E$. The set of points of order $m$ is denoted by

$$E[m] = \{P \in E : [m]P = O\}$$

These points are called points of finite order or torsion points.

It is noted that $E[m]$ forms an additive subgroup of $E$. If the coordinates of points are chosen from a particular field $K$. Then we will represent it as $E(K)[m]$. If we add two points $P$ and $Q$ in $E[m]$ then $P + Q$ is also in $E[m]$. Similarly $-P$ is also in $E[m]$. The group of points of order $m$ has a simple structure if the coordinates of the points are chosen from a large field.

**Proposition 7.1.** Let $m \geq 1$ be an integer.

(a) Let $E$ be an elliptic curve over $\mathbb{Q}$ or $\mathbb{R}$ or $\mathbb{C}$. Then

$$E(\mathbb{C})[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$$

is a product of two cyclic group of order $m$. 

171
(b) Let $E$ be an elliptic curve over $\mathbb{F}_p$, and if $p$ does not divide $m$. Then there exist a value for $k$ such that

$$E(\mathbb{F}_{p^k})[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}, \quad \text{for all } j \geq 1$$

**Remark 7.8.1.** If $l$ is prime and $K$ is any field. Then

$$E(K)[l] = \mathbb{Z}/l\mathbb{Z} \times \mathbb{Z}/l\mathbb{Z}$$

even if $m$ is not prime

$$E(K)[m] = \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$$

We have to consider $E[m]$ as a 2-dimensional vector space over the field having basis $P_1, P_2$. Every point $P$ in $E[m]$ can be represented as a linear combination of $P_1$ and $P_2$, i.e., $P = aP_1 + bP_2$, for a unique choice of coefficients $a, b \in \mathbb{Z}/m\mathbb{Z}$. Finding $a$ and $b$ given $P_1, P_2$ and $P$ is as hard as ECDLP.

### 7.9 Rational Functions and Divisors on Elliptic Curves

Rational functions on an elliptic curve is related to its zeros and poles. Consider a rational function of a single variable. A rational function is a ratio of polynomials.

$$f(X) = \frac{a_0 + a_1X + a_2X^2 + \cdots + a_nX^n}{b_0 + b_1X + b_2X^2 + \cdots + b_nX^n}$$

The polynomial can be factored completely if we allow complex numbers. So the rational function $f(X)$ can be factored as

$$f(X) = \frac{a(X - \alpha_1)^{c_1}(X - \alpha_2)^{c_2} \cdots (X - \alpha_r)^{c_r}}{b(X - \beta_1)^{d_1}(X - \beta_2)^{d_2} \cdots (X - \beta_s)^{d_s}}$$
The numbers $\alpha_1, \alpha_2, \ldots, \alpha_r$ are called zeros of $f(X)$ and the numbers $\beta_1, \beta_2, \ldots, \beta_s$ are called the poles of $f(X)$. The exponents $e_1, e_2, \ldots, e_r$ are the associated multiplicities of zeros and the exponents $d_1, d_2, \ldots, d_s$ are the associated multiplicities of poles. We can keep track of the zeros and poles of $f(X)$ and their multiplicities by defining the divisor of $f(X)$, which is the formal sum.

$$\text{div}(f(X)) = e_1[\alpha_1] + e_2[\alpha_2] + \cdots + e_r[\alpha_r] - d_1[\beta_1] - d_2[\beta_2] - \cdots - d_r[\beta_r]$$

In a similar fashion, if $E$ is an elliptic curve

$$E : Y^2 = X^3 + AX + B$$

and if $f(X,Y)$ is a rational function of two variables, then there are points on $E$ where the numerator of $f$ vanishes and also there are points where the denominator of $f$ vanishes. That is $f$ has zeros and poles on $E$. We can assign multiplicities to the zeros and poles, so $f$ has a divisor.

$$\text{div}(f) = \sum_{P \in E} n_P[P]$$

The coefficients $n_P$ are integers and only finitely many of the $n_P$ are non zero, so $\text{div}(f)$ is a finite sum.

**Example 7.9.1.** Suppose $E$ defined by the cubic equation factors as

$$X^3 + AX + B = (X - \alpha_1)(X - \alpha_2)(X - \alpha_3)$$

Then the points $P_1 = (\alpha_1,0), P_2 = (\alpha_2,0)$ and $P_3 = (\alpha_3,0)$ are points of order 2, i.e., $2P_1 = 2P_2 = 2P_3 = O$. The divisor of $Y$ is equal to

$$\text{div}(Y) = [P_1] + [P_2] + [P_3] - 3[O]$$

173
In general, we can define divisor on $E$ to be any formal sum

$$D = \sum_{P \in E} n_P [P] \quad \text{with } n_P \in \mathbb{Z} \text{ and } n_P = 0 \text{ for all but finitely many } P$$

The degree of a divisor is the sum of its coefficients

$$\deg(D) = \sum_{P \in E} n_P$$

The sum of the divisor is

$$\text{Sum}(D) = \sum_{P \in E} n_P P$$

The following theorem says which divisors are divisors of functions and to what extent the divisor of a function determines the function.

**Theorem 7.9.1.** [99] Let $E$ be an elliptic curve

1. Let $f$ and $f'$ be rational functions on $E$. If $\text{div}(f) = \text{div}(f')$, then there is a non zero constant $c$ such that $f = cf'$

2. Let $D = \sum_{P \in E} n_P [P]$ be a divisor on $E$. Then $D$ is the divisor of a rational function on $E$, if and only if

$$\deg(D) = 0 \quad \text{and} \quad \text{Sum}(D) = \mathcal{O}$$

In particular, if a rational function on $E$ has no zeros or no poles, then it is a constant.

It is noted that, if $P \in E[m]$ is a point of order $m$. Then $m[P] = \mathcal{O}$, so the divisor

$$m[P] - m[\mathcal{O}]$$

satisfies the conditions of the above theorem. Hence there is a rational function $f_P(X,Y)$ on $E$ satisfying

$$\text{div}(f_P) = m[P] - m[\mathcal{O}]$$

174
Example 7.9.2. Consider the case when \( m = 2 \). The points of order 2 has \( Y \) coordinate 0. If \( P = (\alpha, 0) \in E[2] \), then the function \( f_P = X - \alpha \) satisfies

\[
div(X - \alpha) = 2[P] - 2[O]
\]

7.10 Bilinear Pairing on Elliptic Curve

Bilinear Pairing on elliptic curve is an important construct which provides solution to several security problems. There are lot of examples of bilinear pairing in linear algebra. The dot product is a bilinear pairing on the vector space \( \mathbb{R}^n \). The pairing takes two vectors \( v \) and \( w \) from \( \mathbb{R}^n \) and return a number.

\[
\beta(v, w) = v \cdot w = v_1 w_1 + v_2 w_2 + \cdots + v_n w_n
\]

It is linear in the sense that for any vectors \( v_1, v_2, w_1, w_2 \) and any real numbers \( a_1, a_2, b_1, b_2 \), it satisfies the relation

\[
\beta(a_1 v_1 + a_2 v_2, w) = a_1 \beta(v_1, w) + a_2 \beta(v_2, w)
\]

\[
\beta(v, b_1 w_1 + b_2 w_2) = b_1 \beta(v, w_1) + b_2 \beta(v, w_2)
\]

Another bilinear pairing is the determinant map on \( \mathbb{R}^2 \). Thus if \( v = (v_1, v_2) \) and \( w = (w_1, w_2) \), then

\[
\delta(v, w) = \det \begin{pmatrix} v_1 & v_2 \\ w_1 & w_2 \end{pmatrix} = v_1 w_2 - v_2 w_1
\]

is a bilinear map. The determinant map is alternating, which means that, if we switch the vectors, the value changes sign. This also implies that \( \delta(v, v) = 0 \).

\[
\delta(v, w) = -\delta(w, v)
\]
Chapter 7. Elliptic Curve and Pairing

The bilinear pairing on elliptic curve is similar to this. They take two points on an elliptic curve as input and gives as output a number. It satisfies the bilinear property also

\[ \beta(a_1 P_1 + a_2 P_2, Q) = a_1 \beta(P_1, Q) \cdot a_2 \beta(P_2, Q) \]
\[ \beta(Q, b_1 P_1 + b_2 P_2) = b_1 \beta(Q, P_1) \cdot b_2 \beta(Q, P_2) \]

where \( P_1, P_2 \) and \( Q \) are the points on elliptic curve and \( a_1, a_2, b_1, b_2 \) are the elements of the field selected.

Bilinear pairing on elliptic curve have number of important cryptographic applications in practice. Most of these applications need finite fields of prime power order \( \mathbb{F}_{p^k} \).

7.11 The Weil Pairing

The Weil pairing \( (e_m) \) takes as input a pair of points \( P, Q \in E[m] \) and gives as output an \( m^{th} \) root of unity. i.e., \( e_m(P, Q)^m = 1 \). The bilinearity of the Weil pairing is represented by the equations

\[ e_m(P_1 + P_2, Q) = e_m(P_1, Q) e_m(P_2, Q) \]
\[ e_m(P, Q_1 + Q_2) = e_m(P, Q_1) e_m(P, Q_2) \]

**Definition 7.11.1.** Let \( P, Q \) be points of order \( m \) in \( E \) i.e., \( P, Q \in E[m] \). Let \( f_P \) and \( f_Q \) be rational functions on \( E \) satisfying

\[ \text{div}(f_P) = m[P] - m[O] \quad \text{and} \quad \text{div}(f_Q) = m[Q] - m[O] \]

The Weil Pairing of \( P \) and \( Q \) is the quantity

\[ e_m(P, Q) = \frac{f_P(Q + S)}{f_P(S)} \bigg/ \frac{f_Q(P - S)}{f_Q(-S)} \]
where $S$ is any point in $E$ and $S \not\in \{O, P, -Q, P - Q\}$. This ensures that $e_m(P, Q)$ is always defined and is non zero. The value of $e_m(P, Q)$ does not depend on the choice of $f_P, f_Q$ and $S$.

The Weil pairing has many useful properties and are useful for number of cryptographic applications.

**Theorem 7.11.2.**

1. The Weil pairing will return a value which is the $m$th root of unity
   
   $$e_m(P, Q)^m = 1 \quad \text{for all } P, Q \in E[m]$$

2. The Weil pairing will satisfy the bilinear property. i.e., for all $P, P_1, P_2, Q, Q_1, Q_2 \in E[m]$
   
   $$e_m(P_1 + P_2, Q) = e_m(P_1, Q)e_m(P_2, Q)$$
   $$e_m(P, Q_1 + Q_2) = e_m(P, Q_1)e_m(P, Q_2)$$

3. The Weil pairing is alternating, which means that
   
   $$e_m(P, Q) = e_m(Q, P)^{-1} \text{ and } e_m(P, P) = 1 \quad \text{for all } P, Q \in E[m]$$

4. The Weil pairing is non degenerate, which means that
   
   $$e_m(P, Q) = 1 \quad \text{for all } Q \in E[m], \text{ then } P = O$$

If we allow coordinates of points in a sufficiently large field, then $E[m]$ is like a 2-dimensional vector space over the field $\mathbb{Z}/m\mathbb{Z}$. If we choose $P_1, P_2 \in E[m]$ be the basis, then any point $P$ can be written as a linear combination of these basis

$$P = a_PP_1 + b_PP_2 \quad \text{for unique } a_P, b_P \in \mathbb{Z}/m\mathbb{Z}$$

The glory of Weil pairing is that it can be computed very efficiently without expressing $P$ and $Q$ in term of the basis for $E[m]$. Expressing a point in terms of the basis is complicated than solving ECDLP.
7.12 Miller Algorithm to Compute Weil Pairing

Victor Miller [148] developed an algorithm using double-and-add method to efficiently compute the Weil Pairing. The key idea is to rapidly evaluate certain functions with specified divisors.

**Theorem 7.12.1.** Let $E$ be an elliptic curve and $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$ are the non zero points on the curve.

(a) Let $\lambda$ be the slope of the line connecting $P$ and $Q$. If $P = Q$, it is the slope of the tangent at $P$. If the line is vertical $\lambda = \infty$. A function $g_{P,Q}$ on $E$ is defined as follows:

$$
g_{P,Q} = \begin{cases} 
\frac{y_Q - y_P - \lambda(x_P - x_Q)}{x_P + x_Q - \lambda^2} & \text{if } \lambda \neq \infty, \\
 x - x_P & \text{if } \lambda = \infty.
\end{cases}
$$

Then

$$\text{div}(g_{P,Q}) = [P] + [Q] - [P + Q] - [O]$$

(b) Miller’s Algorithm.

Let $m \geq 1$ and write the binary expansion of $m$ as

$$m = m_0 + m_12 + m_22^2 + \ldots + m_{n-1}2^{n-1}$$

with $m_i \in \{0, 1\}$ and $m_{n-1} \neq 0$. The algorithm returns a function $f_P$ whose divisor satisfies

$$\text{div}(f_P) = m[P] - [mP] - (m - 1)[O]$$
Algorithm 7.2: Miller’s Algorithm

\textbf{Input:} An Elliptic Curve point \( P \) of order \( m \)

\textbf{Output:} A function \( f_P \) with \( \text{div}(f_P) = m[P] - m[O] \)

1. Set \( T = P \) and \( f = 1 \)
2. for \( i = n - 2 : 0 \) do
   3. Set \( f = f^2.g_{T,T} \)
   4. Set \( T = 2T \)
   5. if \( m_i = 1 \) then
      6. set \( f = f.g_{T,P} \)
      7. set \( T = T + P \)
   end
9. end
10. Return the value \( f \)

Proof. a) Let \( y = \lambda x + v \) be the line through \( P \) and \( Q \) or the tangent line at \( P \) if \( P = Q \). The line intersect \( E \) at the three points \( P, Q \) and \(- (P + Q)\), so

\[
\text{div}(y - \lambda x - v) = [P] + [Q] + [-P - Q] - 3[O]
\]

vertical lines intersect \( E \) at points and their negatives, so

\[
\text{div}(x - x_{P+Q}) = [P + Q] + [-P - Q] - 2[O]
\]

It is noted that

\[
g_{P,Q} = \frac{y - \lambda x - v}{x - x_{P+Q}} \quad (7.1)
\]

has the divisor

\[
[P] + [Q] - [P + Q] - [O]
\]

According to the Addition theorem \( x_{P+Q} = \lambda^2 - x_P - x_Q \). Let \( y_P = \lambda x_P + v \) so \( v = y_P - \lambda x_P \). Replacing the values of \( v \) and \( x_{P+Q} \) in
equation 7.1 will result
\[
g_{P,Q} = \frac{y - \lambda x - y_P + \lambda x_P}{x - \lambda^2 + x_P + x_Q}
\]
\[
g_{P,Q} = \frac{y - y_P - \lambda(x - x_P)}{x + x_P + x_Q - \lambda^2}
\]
If \( \lambda = \infty \), then \( P + Q = \mathcal{O} \), so \( g_{P+Q} \) have divisor \([P] + [-P] - 2[\mathcal{O}]\).

b) The Miller’s algorithm is similar to double-and-add algorithm. The function \( g_{T,T} \) in step 3 and \( g_{T,P} \) in step 6 have divisors
\[
div(g_{T,T}) = 2[T] - [2T] - [\mathcal{O}] \quad \text{and} \quad div(g_{T,P}) = [T] + [P] - [T + P] - [\mathcal{O}]
\]
Using induction on this relation, it can be proved that \( f_P \) is a function with divisor \( m[P] - m[\mathcal{O}] \)

Let \( P \in E[m] \), then the Miller algorithm can be used to compute a function \( f_P \) with divisor \( m[P] - m[\mathcal{O}] \). If \( R \) is any point on \( E \), then we can compute \( f_P(R) \) by evaluating the functions \( g_{T,T}(R) \) and \( g_{T,P}(R) \) during the execution of the algorithm. It is noted that for computing Weil pairing, we have to evaluate the function at each of the specified point in the given formula
\[
e_m(P, Q) = \frac{f_P(Q + S)}{f_P(S)} \left/ \frac{f_Q(P - S)}{f_Q(-S)} \right.
\]
One can compute \( f_P(Q + S) \) and \( f_P(S) \) simultaneously for efficiency, and similarly for \( f_Q(P - S) \) and \( f_Q(-S) \). Further savings in computations are available using the Tate pairing, which is a variant of the Weil pairing that we discuss next.
7.13 The Tate Pairing

The Weil pairing on elliptic curve is defined over any field. For elliptic curves over finite fields there is another efficient pairing is defined called Tate pairing. It is computationally more efficient than Weil pairing.

**Definition 7.13.1.** Let $E$ be an elliptic curve over $\mathbb{F}_q$ and let $l$ be a prime. If $P$ and $Q$ are the two points on $E(\mathbb{F}_q)$ such that $P \in E(\mathbb{F}_q)[l]$ and $Q \in E(\mathbb{F}_q)$. Choose a rational function $f_P$ on $E$ with

$$\text{div}(f_P) = l[P] - l[O]$$

Then the Tate pairing of $P$ and $Q$ is the quantity

$$\tau(P, Q) = \frac{f_P(Q + S)}{f_P(S)} \in \mathbb{F}_q^*$$

where $S$ is any point in $E(\mathbb{F}_q)$ such that $f_P(Q + S)$ and $f_P(S)$ are defined and is non zero. If $q \equiv 1 \pmod{l}$, then the modified Tate pairing of $P$ and $Q$ to be

$$\hat{\tau}(P, Q) = \tau(P, Q)^{(q-1)/l} = \left(\frac{f_P(Q + S)}{f_P(S)}\right)^{(q-1)/l} \in \mathbb{F}_q^*$$

**Theorem 7.13.2.** Let $E$ be an elliptic curve over $\mathbb{F}_q$ and $l$ be a prime such that

$q \equiv 1 \pmod{l}$ and $E(\mathbb{F}_q)[l] \cong \mathbb{Z}/l\mathbb{Z}$

Then the modified Tate pairing gives a well-defined map

$$\hat{\tau} : E(\mathbb{F}_q)[l] \times E(\mathbb{F}_q)[l] \to \mathbb{F}_q^*$$

The Tate pairing satisfies the following properties:

**Bilinearity**

$$\hat{\tau}(P_1 + P_2, Q) = \hat{\tau}(P_1, Q)\hat{\tau}(P_2, Q) \text{ and}$$

$$\hat{\tau}(P, Q_1 + Q_2) = \hat{\tau}(P, Q_1)\hat{\tau}(P, Q_2)$$
Nondegeneracy

\[ \hat{\tau}(P, P) \text{ is a primitive } l^{th} \text{ root of unity for all nonzero } P \in E(\mathbb{F}_q)[l] \]

( if \( x \) is the primitive \( l^{th} \) root of unity, then \( x^l = 1 \))

Miller’s algorithm can be used to compute the function \( f_P \) and the Tate pairing efficiently. An efficient implementation of Tate pairing is given in [76].

7.14 MOV Algorithm

The Menezes, Okamoto and Vanstone (MOV) algorithm [145] reduces the ECDLP in \( E(\mathbb{F}_p) \) to DLP problem in \( \mathbb{F}_p^* \). Let \( E \) be an elliptic curve over \( \mathbb{F}_p \), and let \( m \geq 1 \) be an integer such that \( p \nmid m \). The curve has \( m^2 \) points of order \( m \), but their coordinates may lie in a larger field. We can define the term embedding degree as follows.

**Definition 7.14.1.** Let \( E \) be an elliptic curve over \( \mathbb{F}_p \) and let \( m \geq 1 \) be an integer with \( p \nmid m \). The embedding degree of \( E \) with respect to \( m \) is the smallest value of \( k \) such that

\[ E(\mathbb{F}_{p^k})[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z} \]

If \( m \) is a large prime, then the embedding degree have following characterization, which is suitable for cryptographic applications.

**Proposition 7.2.** Let \( E \) be an elliptic curve over \( \mathbb{F}_p \) and let \( l \neq p \) be a prime. If \( E(\mathbb{F}_p) \) contains a point of order \( l \), then the embedding degree of \( E \) with respect to \( l \) is given by:

(i) The embedding degree of \( E \) is 1.(This cannot happen if \( l > \sqrt{p} + 1 \)).

(ii) If \( p \equiv 1 \, (\text{mod } l) \), then the embedding degree is \( l \).
(iii) If \( p \not\equiv 1 \pmod{l} \), then the embedding degree is the smallest value of \( k \geq 2 \) such that

\[
p^k \equiv 1 \pmod{l}
\]

The significance of the embedding degree \( k \) is that, Weil pairing can be used to embed ECDLP in \( E(\mathbb{F}_p) \) into the DLP in the field \( \mathbb{F}_{p^k} \).

Let \( E \) be an elliptic curve over \( \mathbb{F}_p \). If \( l > \sqrt{p} + 1 \) be a large prime. Let \( k \) be the embedding degree and the DLP in \( \mathbb{F}_{p^k}^* \) is solvable. Then if \( P,Q \in E(\mathbb{F}_p) \) such that \( Q = nP \), the MOV algorithm can be used to solve ECDLP and find \( n \).

**Algorithm 7.3: MOV Algorithm**

| **Input:** Elliptic Curve points \( P \) and \( Q \) such that \( Q = nP \). |
| **Output:** The solution of ECDLP i.e., \( n \). |
| 1 Compute the number of points \( N = \#E(\mathbb{F}_{p^k}) \). |
| 2 Choose a random point \( T \in E(\mathbb{F}_{p^k}) \) and \( T \notin E(\mathbb{F}_p) \). |
| 3 Compute \( T' = (N/l)T \). If \( T' = O \), go back to step 2, else \( T' \) is a point of order \( l \) and proceed to the next step 4. |
| 4 Compute the Weil pairing values

\[
\alpha = e_l(P, T') \in \mathbb{F}_{p^k}^* \text{ and } \beta = e_l(Q, T') \in \mathbb{F}_{p^k}^*
\]

Solve the DLP for \( \alpha \) and \( \beta \) in \( \mathbb{F}_{p^k}^* \), i.e., find an exponent \( n \) such that \( \beta = \alpha^n \). |
| 5 Since \( Q = nP \), the ECDLP is also solved. |
| 6 Return \( n \). |

**Remark 7.14.1.** There exist polynomial time algorithm to compute the number of points, if \( k \) is not so large. The Weil pairing computation in step 4 can be done quite efficiency using Miller’s algorithm in time proportional to \( \text{log}(p^k) \). The DLP can be solved using the index calculus method which is a sub exponential algorithm and is considerably faster than the collision algorithms such as Pollard’s \( \rho \) method.

183
The point $T'$ constructed is independent of $P$ and they form a basis of

$$E[l] \cong \mathbb{Z}/l\mathbb{Z} \times \mathbb{Z}/l\mathbb{Z}$$

$e_l(P, T')$ is the non trivial $l^{th}$ root of unity in $\mathbb{F}_{p^k}^*$. The linearity of Weil pairing implies that

$$e_l(P, T') = e_l(nP, T') = e_l(P, T')^n = e_l(Q, T')$$

So $n$ solves the ECDLP for $P$ and $Q$.

The practicality of MOV algorithm depends on the size of $k$. If $k$ is large, say $k > (\ln p)^2$, then the MOV algorithm is completely infeasible. However there are certain special curves whose embedding degree is small. An important class of such curves satisfying the property that

$$\#E(\mathbb{F}_p) = p + 1$$

These curves are called super singular curves [75]. They have the embedding degree $k = 2$ and in any case $k \leq 6$.

For example the curve $E : y^2 = x^3 + x$ is super singular for any prime $p \equiv 3 \pmod{4}$ and it has an embedding degree 2 for any $l > \sqrt{p} + 1$. Solving ECDLP in $E(\mathbb{F}_p)$ is no harder than solving DLP in $\mathbb{F}_{p^2}^*$. This means that, it is a poor choice for the applications in cryptography.

There exist another class of elliptic curves over $\mathbb{F}_p$ called anomalous. They have the property $\#E(\mathbb{F}_p) = p$. There exist fast linear time algorithm to solve ECDLP on these curves [184]. So the use of these curves must also be avoided.

The ECDLP is also easy in elliptic curves defined over $\mathbb{F}_{2^m}$, when $m$ is composite. The idea is to transfer the ECDLP in $\mathbb{F}_{2^m}$ to an hyperelliptic curve over a smaller field $\mathbb{F}_{2^k}$, where $k$ divides $m$. 

184
7.15 Modified Weil Pairing and Distortion Maps

In cryptographic application, we want to evaluate $e_m(P, P)$ or $e_m(aP, bP)$. But the Weil pairing is alternating, which means that $e_m(P, P) = 1$, for all $P$. So the direct use of Weil pairing is not useful. Let $P_1 = aP$ and $P_2 = bP$.

$$e_m(P_1, P_2) = e_m(aP, bP) = e_m(P, P)^{ab} = 1$$

One way to get around this is to use an elliptic curve that has a map $\phi : E \rightarrow E$, with the property that $P$ and $\phi(P)$ are independent in $E[m]$. Hence we can evaluate

$$e_m(P_1, P_2) = e_m(P_1, \phi(P_2)) = e_m(aP, b\phi(P)) = e_m(P, \phi(P))^{ab}$$

For cryptographic application, we choose $m$ to be prime.

**Definition 7.15.1.** Let $E$ be an elliptic curve and $l \geq 3$ be prime. Let $P \in E[l]$ be a point of order $l$ and let $\phi : E \rightarrow E$ be a map from $E$ to itself. Then the map $\phi$ is called $l$ distortion map, if it has the following properties.

(i) $\phi(nP) = n\phi(P)$ for all $n \geq 1$.

(ii) The pairing $e_l(P, \phi(P))$ is a primitive $l^{th}$ root of unity. i.e.,

$$e_l(P, \phi(P))^r \equiv 1 \pmod{l}$$

The modified Weil pairing is defined in the following way.

**Definition 7.15.2.** Let $E$ be an elliptic curve. Let $P \in E[l]$ and let $\phi$ be an $l$ distortion map for $P$. The modified Weil pairing $\hat{e}_l$ on $E[l]$ is defined by

$$\hat{e}_l(Q, Q') = e_l(Q, \phi(Q'))$$
For cryptographic applications, the Weil pairing is evaluated at points that are multiple of $P$. The important property of modified Weil pairing is its non degeneracy. If $Q$ and $Q'$ are the multiples of $P$, then

$$\hat{e}_l(Q, Q') = 1 \quad \text{if and only if} \quad Q = \mathcal{O} \text{ or } Q' = \mathcal{O}$$

**Example 7.15.1.** Let us choose an elliptic curve $E : y^2 = x^3 + x$ over the field $\mathbb{F}_p$ with $p \equiv 3 \pmod{4}$. Let $\alpha \in \mathbb{F}_p^2$ satisfying $\alpha^2 = -1$. The map is defined by

$$\phi(x, y) = (-x, \alpha y) \quad \text{and} \quad \phi(\mathcal{O}) = \mathcal{O}$$

Let $l \geq 3$ be a prime and there exist a non zero point $P \in E(\mathbb{F}_p)[l]$. Then $\phi$ is a $l$ distortion map for $P$

$$\hat{e}_l(P, P) = e_l(P, \phi(P))$$

is the primitive $l^{th}$ root of unity. Since $\phi(P)$ in $E(\mathbb{F}_p^2)$, it is a self map. The map $\phi$ respect the addition law on $E$.

$$\phi(P_1 + P_2) = \phi(P_1) + \phi(P_2) \quad \text{for all } P_1, P_2 \in E(\mathbb{F}_p^2)$$

In particular $\phi(nP) = n\phi(P)$ for all $n \geq 1$.

### 7.16 Concluding Remarks

In this section we explored just enough theory of elliptic curve and pairing. These are the building blocks of several secret sharing constructions based on elliptic curve. The secret sharing schemes with enhanced capabilities can be build using elliptic curve and bilinear pairing. Several advantages are also achieved by the use of elliptic curves for building secret sharing techniques. Share verification, cheater identification and cheater detection are the major achievements with less
computational complexity. Cheater detection and identification can be easily achieved with pairing based techniques. The hardness of ECDLP helps in maintaining the security of shares when building multi secret sharing techniques. The security it offers is comparatively high.