5

Elliptic Curve Cryptography and Point Counting Algorithms

Hailiza Kamarulhaili and Liew Khang Jie
School of Mathematical Sciences, Universiti Sains Malaysia, Minden, Penang Malaysia

1. Introduction

Elliptic curves cryptography was introduced independently by Victor Miller (Miller, 1986) and Neal Koblitz (Koblitz, 1987) in 1985. At that time elliptic curve cryptography was not actually seen as a promising cryptographic technique. As time progress and further research and intensive development done especially on the implementation side, elliptic curve cryptography is now being implemented widely. Elliptic curves cryptography offers smaller key size, bandwidth savings and faster in implementations when compared to the RSA (Rivest-Shamir-Adleman) cryptography which based its security on the integer factorization problem. The most interesting feature of the elliptic curves is the group structure of the points generated by the curves, where points on the elliptic curves form a group. The security of elliptic curves cryptography relies on the elliptic curves discrete logarithm problem. The elliptic curve discrete logarithm problem is analogous to the ordinary algebraic discrete logarithm problem, $l = g^x$, where given the l and g, it is infeasible to compute the x. Elliptic curve discrete logarithm problem deals with solving for n the relation $P = nG$. Given the point P and the point G, then it is very hard to find the integer n. To implement the discrete logarithm problem in elliptic curve cryptography, the main task is to compute the order of group of the curves or in other words the number of points on the curve. Computation to find the number of points on a curve, has given rise to several point counting algorithms. The Schoof and the SEA (Schoof-Elkies-Atkin) point counting algorithms will be part of the discussion in this chapter. This chapter is organized as follows: Section 2, gives some preliminaries on elliptic curves, and in section 3, elliptic curve discrete logarithm problem is discussed. Some relevant issues on elliptic curve cryptography is discussed in section 4, in which the Diffie-Hellman key exchange scheme, ElGamal elliptic curve cryptosystem and elliptic curve digital signature scheme are discussed here accompanied with some examples. Section 5 discussed the two point counting algorithms, Schoof algorithm and the SEA (Schoof-Elkies-Atkin) algorithm. Following the discussion in section 5, section 6 summaries some similarities and the differences between these two algorithms. Section 7 gives some brief literature on these two point counting algorithms. Finally, section 8 is the concluding remarks for this chapter.

2. Elliptic curves

Elliptic curves obtained their name from their relation to elliptic integrals that arise from the computation of the arc length of ellipses (Lawrence & Wade, 2006). Elliptic curves are

different from ellipses and have much more interesting properties when compared to ellipses. An elliptic curve is simply the collection of points in *x*-*y* plane that satisfy an equation $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, and this equation could either be defined on real, rational, complex or finite field. This equation is called the Weierstrass equation.

Definition 2.1: An elliptic curve *E*, defined over a field *K* is given by the Weierstrass equation:

$$E:\ y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\ ,\ \text{where}\ \ a_1, a_2, a_3, a_4, a_6 \in K \tag{1}$$

In other words, let *K* be any field, then we assume $a_1, a_2, a_3, a_4, a_6 \in K$ and the set of *K*-rational points:

$$E(K) = \{(x, y) \mid x, y \in K, y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\}.$$

If one is working with characteristic, char $(K) \neq 2,3$, then admissible changes of variables will transform the above equation (1) into the following form:

$$y^2 = x^3 + ax + b\ \ \text{where}\ \ a, b \in K \tag{2}$$

But when one works with $char(K) = 2$ or 3 , then the general form of equation is given by (3) and (4) respectively.

$$y^2 + xy = x^3 + a_2x^2 + a_6 \tag{3}$$

$$y^2 = x^3 + a_2x^2 + a_6 \tag{4}$$

2.1 Case for real numbers

This case allows us to work with graphs of *E*. The graph of *E* has two possible forms, whether the cubic polynomial has only one real root or three real roots. Now, we consider the following examples. Take the equations $y^2 = x(x+1)(x-1)$ and $y^2 = x^3 + 73$. The graphs are as follows:

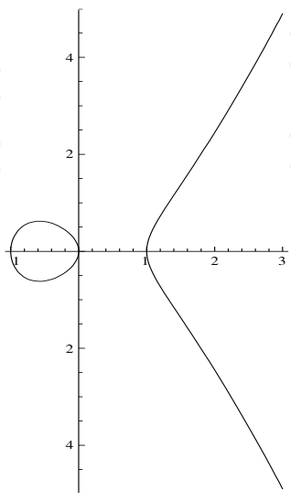


Fig. 1.1. $y^2 = x(x+1)(x-1)$

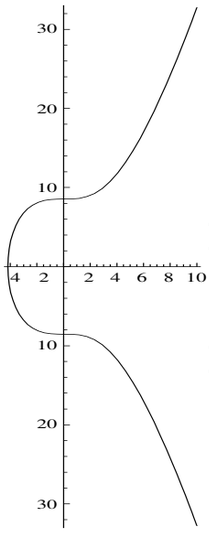


Fig. 1.2. $y^2 = x^3 + 73$.

Looking at the curves, how do you create an algebraic structure from something like this. Basically, one needs to figure out how to find a way to define addition of two points that lie on the curve such that the sum is another point which is also on the curve. If this could be done, together with an identity element, O_∞ , group structure can be constructed from points on the curves. The following are some formulas for points operations on the curves which is defined by the equation (2).

1. $P + O_\infty = P$, for all points P.
2. $-P = O_\infty - (P)$
3. The opposite point, $-P = (x, -y)$
4. $P = (x_1, y_1)$ & $Q = (x_2, y_2)$, then $P + Q = R = (x_3, y_3)$, with

$$x_3 = m^2 - x_1 - x_2,$$
$$y_3 = m(x_1 - x_3) - y_1,$$
$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{if} \quad Q \neq \pm P$$

or

$$m = \frac{3x_1^2 + a}{2y_1} \quad \text{if} \quad P = Q.$$

It can be shown that the addition law is associative, that is

$$(P + Q) + R = P + (Q + R).$$

It is also commutative,

$$P + Q = Q + P .$$

When several points are added, it does not matter in what order the points are added or how they are grouped together. Technically speaking, the points on the curve, E form an abelian group. The point O_∞ is the identity element of this group.

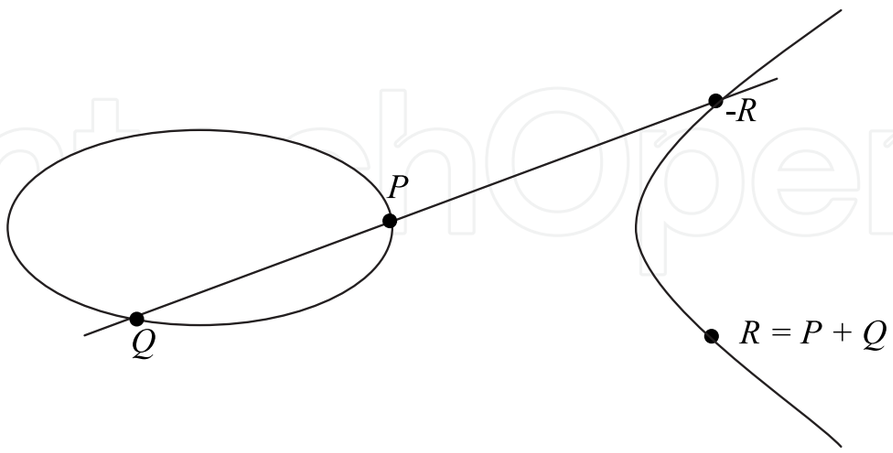


Fig. 1.3. Addition of elliptic curve points over a real number curve

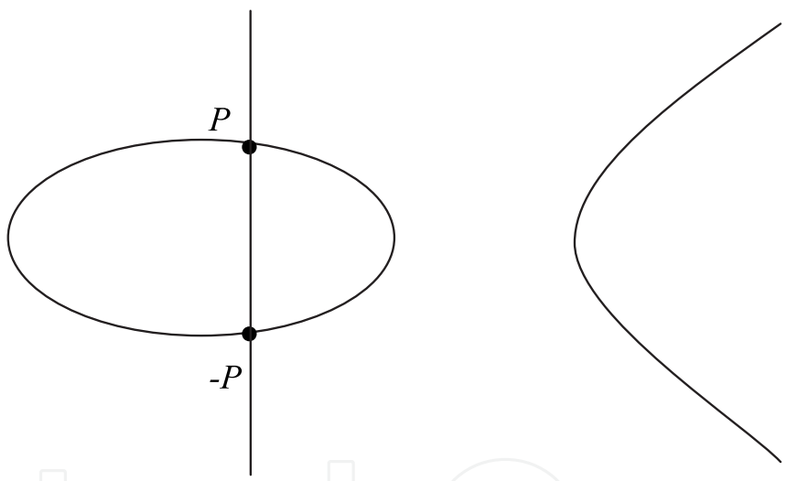


Fig. 1.4. Arbitrary points P and $-P$

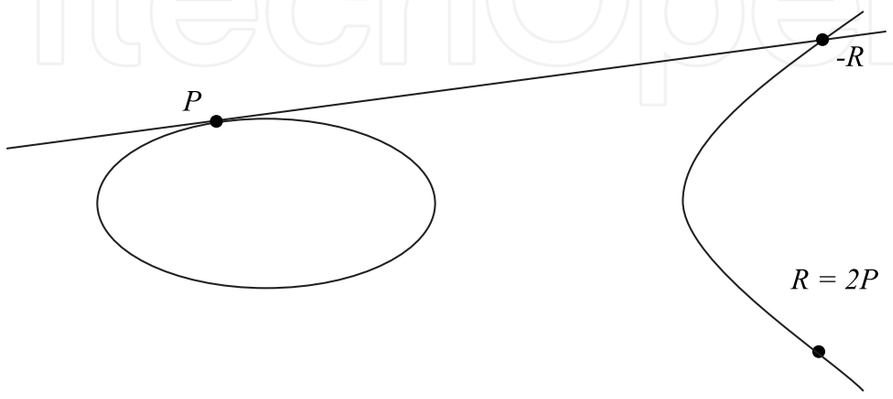


Fig. 1.5. Addition of a point to itself (point doubling)

2.2 Case for integer mod *p* (prime field)

The operations of points on elliptic curves indicated in the previous section are fascinating and it is applicable to the area of cryptography. It so happen that similar formulas work if real numbers are replaced with finite field. An elliptic curve defined over prime field is cryptographically good if the curve is non-singular. This happens when the discriminant, $-16(4a^3 + 27b^2) \neq 0$. That means, the polynomial $x^3 + ax + b$ has no multiple roots.

Now define an elliptic curve mod p, where p is a prime. For the rest of this section several examples are shown to exhibit its cryptographic use.

Example 2.1: Let E be given by $y^2 \equiv x^3 + 2x - 1 \pmod 5$. First of all, compute and list all the points on the curve by letting x run through the values 0, 1, 2, 3, 4 and solve for y. Substitute each of these into the equation and find the values of y that solve the equation.

$$
\begin{aligned}
x \equiv 0 \Rightarrow y^2 \equiv -1 \equiv 4 \quad &\Rightarrow \quad y \equiv 2,3 \pmod 5 \\
x \equiv 1 \Rightarrow y^2 \equiv 2 \quad &\Rightarrow \quad \text{no solution} \\
x \equiv 2 \Rightarrow y^2 \equiv 11 \equiv 1 \equiv 16 \quad &\Rightarrow \quad y \equiv 1,4 \pmod 5 \\
x \equiv 3 \Rightarrow y^2 \equiv 32 \equiv 2 \quad &\Rightarrow \quad \text{no solution} \\
x \equiv 4 \Rightarrow y^2 \equiv 71 \equiv 1 \equiv 16 \quad &\Rightarrow \quad y \equiv 1,4 \pmod 5
\end{aligned}
$$

Therefore, yield the following points along with point at infinity, the identity element:

$$(0,2), (0,3), (2,1), (2,4), (4,1), (4,4), (\infty,\infty)$$

Elliptic curves mod p generates finite sets of points and it is these elliptic curves that are useful in cryptography. For cryptographic purposes, the polynomial $x^3 + ax + b$ is assumed not to have multiple roots, as it will lead to weak curves and vulnerable to attack. Computation of points on elliptic curve can also be obtained by using the *Mathematica* software. Now we demonstrate how it can be done. First we need to choose the base point *G*, and the coefficient *a*. Then choose the coefficient *b*, so that *G* lies on the curve $y^2 \equiv x^3 + ax + b \pmod 5$. Now say the point *G* = (1, 3) and choose *a* = 2. Then substitute this into the equation, give the value of *b* = 1. Thus we have $y^2 \equiv x^3 + 2x + 1 \pmod 5$. The following points are generated using the *Mathematica* programming software. The command **multsell** is used to generate points from the curve and was fully written by Lawrence Washington (Lawrence & Wade, 2006). The following are the points generated using the **multsell** command. Thus the following points are generated.

$$(1,3),(3,2),(0,4),(0,1),(3,3),(1,2),(\infty,\infty)$$

2.2.1 Points addition and doubling on elliptic curves

As it was shown earlier in the formulations of points on an elliptic curve, adding points on elliptic curve is not the same as adding points in the plane. Scalar multiplication of a point on the curve for which we have say, *mP* with *m* = 2185, will be evaluated as 2(2(2(2(2(2(2(2(2(2(2*P*))) + *P*))))+ *P*))) + *P*. This is called doubling operation. The

following examples show us how addition and doubling operation exactly works using the formulation in section 2.1.

Example 2.2 (point addition) : Suppose E is defined by $y^2 = x^3 + 2x + 1 \pmod 5$. Now add the point (1, 2) and the point (3, 2). The slope $m \equiv \frac{2-2}{3-2} \equiv 0 \pmod 5$. Then, we have the following formulas to obtain the third point on the curve.

$$x_3 = -1 - 3 = -4 \equiv 1 \pmod 5$$
$$y_3 = -2 \equiv 3 \pmod 5$$

This means that (1, 2) + (3, 2) = (1, 3), which is also on the curve. This can be verified using the *Mathematica* function, **addell** which was also developed by Lawrence C. Washington (Lawrence & Wade, 2006).

Example 2.3 (point doubling): Using the same E as in example 1.2, compute $2P = P + P$, where P= (1, 3). This operation is called doubling.

$$m = \frac{3(1)+2}{2(3)} = \frac{5}{6} \equiv 5 \cdot 6 \equiv 0 \pmod 5$$
$$\therefore m \equiv 0 \pmod 5$$

$$x_3 = -1 - 1 = -2 \equiv 3 \pmod 5$$
$$y_3 = -3 \equiv 2 \pmod 5$$

Thus we have $x_3 = 3, y_3 = 2$. Hence $(1,3) + (1,3) = (3,2)$. This also can be verified using the *Mathematica* command, **addell**. For the ordinary scalar multiplication, say, 3P, is evaluated as 2P + P.

3. Elliptic curve discrete logarithm problem

The term, elliptic curve discrete logarithm problem (ECDLP) comes from the classical discrete logarithm problem, $x \equiv g^k \pmod p$, where we want to find k. In the context of elliptic curve, suppose that the points P, Q on an elliptic curve are made known and $Q = kP$ for some k, then find the k. The difficulty of finding the k is what makes the elliptic curves an area which is cryptographically worth exploring for. In other words, elliptic curves cryptosystem rely its security on the difficulty of the discrete logarithm problem and the available efficient algorithms that can solve the discrete logarithm problem.

Solving the elliptic curve discrete logarithm problem is very hard and until now there is no good and efficient algorithm available to solve the problem. Nevertheless there are a few algorithms being widely discussed, which is popular amongst the cryptanalysts. They are analog of Pohlig-Hellman attack, index calculus attack and baby step-giant step attack. The baby-step giant-step attack on discrete logarithm problem works for elliptic curves although it requires too much memory to be practical. Generally speaking, there is no algorithm available to solve the discrete logarithm problem in sub-exponential time.

4. Elliptic curve cryptography

More than twenty years ago, when elliptic curve cryptography was first introduced independently by

Neal Koblitz and Victor Miller, researchers never thought that elliptic curve cryptography could be implemented efficiently and securely. During those times the arithmetic operations on elliptic curves were difficult to perform. The arithmetic on the elliptic curves was not very efficient and it was only meant for academic interest. Since then, a great deal of effort has been put on the study of elliptic curve and its implementation in cryptography. By the late 1990s the implementations were ten times more efficient and this has made the elliptic curves cryptography as a challenge to the RSA (Rivest- Shamir-Adleman) cryptography.

In recent years, the bit length for secure RSA use has increased and this has increased the processing load on applications using RSA. This is due to the development of the integer factorization algorithms which runs in sub-exponential time and as a result, RSA had to choose a very large key for it to sustain the intractability of the system, where as the elliptic curves cryptosystem require fewer bits or shorter key lengths for the same security level, since the security of the elliptic curve cryptography relies on the discrete logarithm problem and the best known algorithm to solve those problems is fully exponential time. Thus reduction in the time, cost as well as the size or bandwidth and memory requirements, which is crucial factor in some applications such as designs of smart cards, where both memory and processing power are limited but requiring high security. For an example, 160 bits in elliptic curve cryptosystem is around 1024 bits in RSA cryptosystem. Nowadays, elliptic curve cryptosystem is one of the important components in Microsoft Windows, email applications, bank cards and in mobile phones.

As it was mentioned earlier that elliptic curves cryptosystem based its security on the hardness of the discrete logarithm problem. One of the most important aspects in elliptic curve cryptosystem is choosing the right curve that preserved the hardness of discrete logarithm problem. One way to ensure this is to avoid singular curves as the discrete logarithm problem for these types of curves can reduce the hardness of the discrete logarithm problem. The arithmetic on these curves can be much faster over these curves and this is due to the fact that several terms vanished and these types of curves are considered weak and the system will no longer be intractable. Therefore, as mentioned earlier in the previous section, elliptic curves suitable for cryptographic use are of type non-singular curves.

4.1 Embedding plaintext on an elliptic curve

Before messages can be encrypted, those messages need to be embedded on the points of the elliptic curve (Lawrence & Wade, 2006). The embedding process encoded the message m, which is already in a number form, as a point on the curve. Let K be a large positive integer so that a failure rate of $1/2^K$ is acceptable in the decoding process, where $K \in Z$. Assume now that m satisfies $(m+1)K < p$. The message m is presented by a number $x = mK + j$, where K is an integer and $0 \le j < K$. For $j = 0,1,2,....,K-1$, compute $x^3 + ax + b \pmod{p}$ and calculate the square root of it. If there is a square root y, then embedded point, $P_m = (x, y)$. Otherwise, increase the j by one and again compute the new x.

Repeat this step until either the square root is found or $j = K$. For the case where j equals K, the mapping of the message to a point failed. In order to recover the message from the embedded point, $P_m = (x, y)$. m can be recovered by computing $\lfloor x / K \rfloor$. Once the messages have been encoded as points on an elliptic curve, then those points can be manipulated arithmetically to hide away those messages. This process is called encryption process. The reverse of the encryption process is called decryption process. There are three versions of classical algorithms, where arithmetic of elliptic curves is being adopted. They are the elliptic curve Diffie-Hellman key exchange, ElGamal elliptic curve cryptosystem and ElGamal elliptic curve digital signature algorithm.

4.2 Elliptic curve diffie-hellman key exchange

Elliptic curve Diffie-Hellman key exchange was first introduced by Diffie and Hellman in the year 1976 (Hellman, 1976). Now we exhibit the implementation of elliptic curve Diffie-Hellman key exchange. Alice and Bob want to exchange a key. Thus, they agreed on a public point generator or the base point G on an elliptic curve $y^2 \equiv x^3 + ax + b (\mathrm{mod}\, p)$. Now choose p = 7211 and a = 1 and the point G = (3, 5). This gives b = 7206. Alice chooses a random integer k_A = 12 and Bob chooses random integer k_B = 23. Alice and Bob keep these private to themselves but publish the $k_A G$ and $k_B G$. In this case we have

$$k_A G = (1794, 6375) \text{ and } k_B G = (3861, 1242).$$

Alice now takes $k_B G$ and multiples by k_A to get the:

$$k_A (k_B G) = 12(3861, 1242) = (1472, 2098).$$

Similarly, Bob takes $k_A G$ and multiples by k_B to get the key:

$$k_B (k_A G) = 23(1794, 6375) = (1472, 2098).$$

Notice that Alice and Bob have the same key.

4.3 Elliptic curve Elgamal cryptosystem

Assuming we have a situation where there are two parties communicating through an insecure channel. The communication is between Alice and Bob. The following example exhibits the use of elliptic curves to encrypt and decrypt messages.

Example 4.1: Firstly, we must generate a curve. Choose the prime p = 8831, the point $G = (x, y) = (3,7)$ and a = 1. To make G lie on the curve $y^2 \equiv x^3 + ax + b (\mathrm{mod}\, p)$, we then obtain b = 19. Alice has a message, represented as a point $P_m = (5, 1743)$ and she wants to send it to Bob. Bob has chosen a random number $a_b = 5$ and published the point $a_b G = (7335, 7164)$. Alice then chooses a random number k = 4. She sends Bob $kG = (254, 2386)$ and $P_m + k(a_b G) = (269, 1803)$. Bob then first calculate $a_b kG = 5(254, 2386) = (4217, 7788)$. Bob then subtract this from $(269, 1803)$:

$$(269,1803)-(4217,7788)=(269,1803)+(4217,-7788)=(5,1743)$$

Now Bob recovered the message $P_m=(5,1743)$ that Alice sent.

4.4 ElGamal elliptic curve digital signature algorithm

A digital signature is an electronic analogue of a hand written signature that allows a receiver to convince a third party that the message is in fact originated from the sender. ElGamal elliptic curve digital signature algorithm is an analogue to the digital signature algorithm proposed earlier by ElGamal in 1985 where some modifications were done to deal with points on an elliptic curve.

Now suppose that Alice wants to sign a message *m*. assuming that *m* is an integer, Alice fixes an elliptic curve $E(\mathrm{mod}\, p)$, where *p* is a large prime and a point *A* on *E*. We assume that the number of points *n* on *E* has been calculated and $0 \le m < n$. Alice also has to choose a private integer *a* and compute *B* = *aA*. The prime *p*, the curve *E*, the integer *n*, and the points *A* and *B* are made public. To sign the message *m*, Alice does the following procedure:

1. Alice chooses a random integer *k* with $1 \le k < n$ and gcd (*k*, *n*) = 1, and computes *R* = *kA* = (*x*, *y*),
2. Now, Alice computes $s \equiv k^{-1}(m-ax)(\mathrm{mod}\, n)$ and
3. Sends the signed message (*m*, *R*, *s*) to Bob.

Note that *R* is a point on *E*, and *m* and *s* are integers. Next, Bob verifies the signature as follows:

1. Bob now downloads Alice's public information *p*, *E*, *n*, *A*, *B*, and
2. Computes $V_1 = xB + sR$ and $V_2 = mA$.
3. Declares the signature valid if $V_1 = V_2$.

We can verify that the verification procedure works because we have the following:

$$V_1 = xB + sR = xaA + k^{-1}(m-ax)(kA) = xaA + (m-ax)A = mA = V_2$$

5. Point counting for *E* (mod *p*)

Let $E: y^2 = x^3 + bx + c(\mathrm{mod}\, p)$ be an elliptic curve. Then the number of points on *E* denoted as #$E(F_p)$, satisfies Hasse's theorem (Jacobson & Hammer, 2009),(Lawrence & Wade,2006). According to Hasse's theorem, the number of points on *E*, #$E(F_p)$, satisfy the following inequality.

$$p+1-2\sqrt{p} \le \#E\left(F_p\right) \le p+1+2\sqrt{p}$$

Number of points on the curve *E* is called the order of the curve. The order of a point is defined by the number of times the point added to itself until the infinity is obtained. The order of any point on the curve *E*, will divide the order of the curve *E*. If the order of the curve has many factors or smooth, then this curve is not cryptographically good. For

cryptography, it is best if the order of the curve is a large prime number. Generally finding order of a curve is not trivial. In a situation where $p \geq 5$ is a prime, for small p, points can be listed by letting $x = 0,1,2,...,p-1$ and seeing when $x^3 + ax + b$ is a square mod p. When p is large, it is infeasible to count the points on the curve by listing them. There are several algorithms that can deal with this problem, They are Schoof's algorithm and Schoof-Elkies-Atkin (SEA) algorithm (Lawrence &Wade, 2006). In principal, there are approximately p points on the curve E and inclusive of the point at infinity, a total of $p + 1$ points is expected to be on the curve. The order of a curve is called 'smooth' if the order of the curve is divisible by many small factors, where this can brings point multiplications to identity (point at infinity). The type of curve which is desirable is of type 'non-smooth' order, where the order of the curve is divisible by a large prime number. The Schoof-Elkies-Atkin point counting method has become sufficiently efficient to find cryptographic curves of prime order over F_p with heuristic time $O(\log^6 p)$. In the next section, we will discuss the two counting point algorithms, the Schoof counting point algorithm and the Schoof-Elkies-Atkins counting point algorithm.

5.1 Schoof and Schoof-Elkies-Atkin (SEA) point counting algorithms

To determine the $\#E(F_p)$, one needs to compute $z = y^2 = x^3 + ax + b$ for each x in F_p and then test if z has a square root in F_p. If there exists $y \in F_p$ such that $y^2 = z$, then we have $2p$ + (a point of infinity) that is $2p + 1$ elements in the group because each x value will produce two values of y. However, according to the theorem of finite fields, there is around ½ of the non-zero elements of F_p are quadratic residues. So, there is approximately $p + 1$ number of points. There are a few point counting algorithms and in this section, we focus only on two point counting methods. They are Schoof and Schoof-Elkies-Atkin (SEA) point counting algorithms. In this chapter, we will describe the two algorithms in a brief manner. Readers are required to have some backgrounds in number theory and algebraic geometry. For more details on arithmetic of elliptic curves, one needs to refer to (Silverman, 1986) and for the introduction on Schoof algorithm, refer to (Schoof, 1985).

5.1.1 Schoof's algorithm

René Schoof (Schoof, 1985) had introduced a deterministic polynomial time algorithm to compute the number of F_p-points of elliptic curve defined over a finite field F_p which was given by Weierstrass form in (2). Schoof algorithm has managed to compute the group order of over 200 digits. In the Schoof algorithm, the characteristic polynomial of Frobenius endomorphism is critical to the development of Schoof's algorithm. Another crucial part in this algorithm is to compute the division polynomials in order to carry out the computation of the order of the group of elliptic curve. If the division polynomials have low degree, then the division polynomials is said to be efficiently computable.

Let E be an elliptic curve defined over F_p denoted as E/F_p, where F_p is a prime field of characteristic $p > 3$. Define the Frobenius endomorphism ϕ_p as the following:

$$\phi_p : E(\overline{F}_p) \to E(\overline{F}_p)$$
$$(x, y) \mapsto (x^p, y^p)$$

The Frobenius map or endomorphism ϕ_p satisfies the characteristic equation (5)

$$\phi_p{}^2 - t\phi_p + p = 0, \forall P \in (\overline{F}_p) \tag{5}$$

where $\overline{F}_p$ is the algebraic closure of the prime field F_p. Let t is the trace of Frobenius endomorphism, then the number of points, $\#E\left(F_p\right)$ is given in (6) as follows:

$$\#E\left(F_p\right) = p + 1 - t, |t| \le 2\sqrt{p} \tag{6}$$

Obviously from equation (5), we have for all points, $P = (x, y) \in E(\overline{F}_p)$ satisfying the following equation (7):

$$(x^{p^2}, y^{p^2}) + p(x, y) = t(x^p, y^p) \tag{7}$$

where scalar multiplication by p or t signifies adding a point to itself p or t times respectively. For $(x, y) \in \overline{E}[l]$, where $E[l] = \{P = (x, y) \in E(\overline{F_p}) \mid [l]P = O_\infty\}$, here each $P \in E[l]$ is called l–torsion point. If $t(x^p, y^p) \equiv \overline{t}(x^p, y^p)$ where $\overline{t}$ is t mod l and $\overline{p}$ known as p mod l where l is a prime. Now, the equation of (7) is reduced as following:

$$(x^{p^2}, y^{p^2}) + \overline{p}(x, y) = \overline{t}(x^p, y^p)$$

To determine t (mod l) for primes $l > 2$, we need to compute the division polynomials.

Definition 5.1.1 (Division Polynomial)

Division polynomial (McGee, 2006) is a sequence of polynomials in $\psi_m \in \mathbb{Z}[x, y, a, b]$ and goes to zero on points of particular order. Let E be the elliptic curve given by (2). The division polynomials $\psi_m(x, y) = 0$ if and only if $(x, y) \in E[n]$. These polynomials are defined recursively as follows (Schoof, 1985):

$\Psi_{-1} = -1$

$\Psi_0 = 0$

$\Psi_1 = 1$

$\Psi_2 = 2y$

$\Psi_3 = 3x^4 + 6ax^2 + 12bx - a^2$

$\Psi_4 = 4y\,(x^6 + 5xa^4 + -20bx^3 - 5a^2x^2 - 4abx - 8b^2 - a^3)$

$\Psi_{2m} = \Psi_m\,(\Psi_{m+2}\,\Psi^2_{m-1} - \Psi_{m-2}\,\Psi^2_{m+1}) \,/\, 2y \qquad m \in \mathbb{Z},\ m \ge 3$

$\Psi_{2m+1} = \Psi_{m+2}\,\Psi^3_m - \Psi^3_{m+1}\,\Psi_{m-1} \qquad m \in \mathbb{Z},\ m \ge 2$

For simplicity, the polynomials are suppressed to $\Psi_{n.}$ which is called the n^{th} division polynomial.

Let us derive the $\Psi_3 = 3x^4 + 6ax^2 + 12bx - a^2$. In division polynomial Ψ_3, we must have a point $P = (x, y) \in E[3]$ which is a point with order 3 such that $[3]P = \infty$. Therefore, we have $2P = -P$ and we know the x- coordinate for point $2P$ and P is the same. The formula for the x-coordinate in $2P$ is given in the earlier section.

$$\begin{aligned} x &= \lambda^2 - 2x \\ &= \left(\frac{3x^2 + a}{2y}\right)^2 - 2x \\ &= \left(\frac{9x^4 + 6ax^2 + a^2}{4y^2}\right) - 2x \end{aligned}$$

$$\begin{aligned} &3x(4y^2) = 9x^4 + 6ax^2 + a^2 \\ &12xy^2 = 9x^4 + 6ax^2 + a^2 \\ &12x(x^3 + ax + b) = 9x^4 + 6ax^2 + a^2 \\ &\psi_3 = 12x^4 + 12ax^2 + 12bx - 9x^4 - 6ax^2 - a^2 = 0 \\ &\therefore \psi_3 = 3x^4 + 6ax^2 + 12bx - a^2 \end{aligned}$$

We can replace y^2 by $(x^3 + ax + b)$ to eliminate the y term. The polynomial $f_n(x) \in F_p[x]$ is defined as follows:

$$f_n(x) = \psi_n(x, y) \text{ if } n \text{ is odd}$$

$$f_n(x) = \frac{\psi_n(x, y)}{y} \text{ if } n \text{ is even}$$

If n is odd, then the degree of $f_n(x)$ is $\frac{n^2 - 1}{2}$ whereas if n is even, then the degree of $f_n(x)$ is $\frac{n^2 - 4}{2}$.

The following proposition shows point additions relates to the division polynomials.

Proposition 5.1.1

Let $(x, y) \in E(\bar{F}_p)$, with $\bar{F}_p$, the algebraic closure of F_p. Let $n \in \mathbb{Z}$, then for $[n]P = P + P + P + \ldots + P$ is given by

$$[n]P = \left(x - \frac{\psi_{n-1}\psi_{n+1}}{\psi_n^2}, \frac{\psi_{n+2}\psi_{n-1}^2 - \psi_{n-2}\psi_{n+1}^2}{4y\psi_n^3}\right)$$

5.1.1.1 Computation of number of points, #E(F_p) using Schoof's algorithm

Here, we present briefly the Schoof's algorithm. For E as defined in (2) over F_p and the Hasse's theorem,

$$\#E(F_p) = p + 1 - t \text{ where } |t| \leq 2\sqrt{p}\,.$$

Input : Elliptic curve $y^2 = x^3 + ax + b$ over prime field F_p.

Output: Number of points, $\#E(F_p)$.

1. Create a set of small primes not equal to the char $(F_p) = p$,

$$S = \{l_1, l_2, ..., l_L\}, = \{2, 3, 5, 7, 11, \ldots l_L\} \text{ such that } \prod_{i=1}^{L} l_i > \left\lceil 4\sqrt{p} \right\rceil.$$

a. For case when the prime $l = 2$:
2. gcd $(x^3 + ax + b, x^p - x) \neq 1$, then $t \equiv 0 \pmod 2$, else $t \equiv 1 \pmod 2$.

This is to test whether E has point of order 2, $(x, 0) \in E[2]$ or precisely roots of E.

If gcd $(x^3 + ax + b, x^p - x)$ = gcd$(x^3 + ax + b, x_p - x) = 1$, then $x^3 + ax + b$ has no root in F_p, else it has at least one such root.

3. To test whether which case is to be used, we have to compute the following relation:

$$\gcd\left((x^{p^2} - x)\psi_{p_l}^2 - \psi_{p_l-1}\psi_{p_l+1} (\mathrm{mod}\, \psi_l, p), \psi_l\right)$$

If the gcd = 1 , proceed to (B), else proceed to (C) .

b. For the case when $(x^{p^2}, y^{p^2}) \neq \pm p_l(x, y)$
4. For each $l \in S$, compute $p_l \equiv p \pmod l$
5. For case $(x^{p^2}, y^{p^2}) \neq \pm p_l(x, y)$
6. Compute $(x', y') = (x^{p^2}, y^{p^2}) + p_l(x, y) \neq \infty$
7. For each $1 \leq \tau \leq \dfrac{l-1}{2}$, compute the x-coordinate, x_τ of $\tau(x, y) = (x_\tau, y_\tau)$
8. If $x' - x_\tau \neq 0 (\mathrm{mod}\, \psi_l)$ then try next τ, else compute y' and y_τ.
9. If $\dfrac{y' - y_\tau}{y} \equiv 0 (\mathrm{mod}\, \psi_l)$, then $t \equiv \tau (\mathrm{mod}\, l)$, else $t \equiv -\tau (\mathrm{mod}\, l)$
10. If all values $1 \leq \tau \leq \dfrac{l-1}{2}$ fail, then proceed to case (C).

c. For the case when $(x^{p^2}, y^{p^2}) = \pm p_l(x, y)$
11. Compute w such that $w^2 \equiv p \pmod l$
12. If w^2 does not exist, then $t \equiv 0 \pmod l$, else
13. Compute $(x^p, y^p) = \pm w(x, y) = \pm(x_w, y_w) = (x_w, \pm y_w)$

14. If gcd(numerator($(x^p - x_w)$, ψ_l) = 1, then $t \equiv 0 (\mathrm{mod}\, l)$
15. Else compute gcd(numerator($(y^p - y_w) / y), \psi_l$)
16. If gcd(numerator($(y^p - y_w) / y), \psi_l$) = 1, then $t \equiv -2w (\mathrm{mod}\, l)$, else $t \equiv 2w (\mathrm{mod}\, l)$

Recover t via Chinese Remainder Theorem (CRT)

17. At this point we have computed t (mod l) for any $l \in S$.
18. $T \equiv t$ (mod N) where $N = \prod_{i=1}^{L} l_i$.
19. If T is in Hasse's bounds, then $t = T$, else $t \equiv -T$ (mod N)
20. $\#E(F_p) = p + 1 - t$.

5.1.2 Schoof-Elkies-Atkin (SEA) algorithm

Schoof's algorithm is not practical because of the exponential growth in the degree of the division polynomial and hence it is not suitable for cryptographic purposes. Atkin and Elkies has improved the Schoof's algorithm by analyzing the method to restrict the characteristic polynomial of elliptic curve such that

$$\chi(\phi_p) = \phi_p^2 - t\phi_p + p \tag{8}$$

where the Frobenius splits over F_l. The discussion will follow the literature found in (Cohen et al., 2006). In 1988, Atkin devised an algorithm to the order of ϕ_p in projective general linear group dimension 2 of F_l, whereas Elkies in 1991 introduced a mean to replace the division polynomial which has degree $(l^2 - 1)/2$ by a kernel polynomial with degree $(l-1)/2$ in Elkies prime procedures. To differentiate between the Elkies prime and Atkin prime, one can calculate the discriminant from (8), so we get $\Delta = t^2 - 4p$. If Δ is a square then the prime l is Elkies prime, else it is Atkin prime. However, we need to classify the primes at the beginning stage and there is no information of t. Therefore this method is not suitable. However, Atkin proved that l-modular polynomial, $\Phi_l(x, y) \in \mathbb{Z}[x, y]$ can be used to differentiate the prime at the early stage of SEA algorithm. SEA algorithm is one of the fastest algorithms for counting the number of points on E over a large prime field. The following part of this section follows the text from (Chen, 2008), (Cohen et al., 2006), and (Galin, 2007).

5.1.2.1 Modular polynomial

Modular polynomial comes from the theory of modular form and the interpretation of elliptic curves over the complex field as lattices. A moderately comprehensive development of the theory can be found in (Silverman, 1986). Before we proceed with the SEA algorithm, we know the modular polynomial. The detail proof of this theorem can be obtained in (Cox, 1989). These polynomials will be used in Elkies and Atkin procedures.

Theorem 5.1.2.1 (modular polynomial)

Let m be a positive integer.

i. $\Phi_m(x,y) \in \mathbb{Z}[x,y]$

ii. $\Phi_m(x,y)$ is irreducible when regarded as polynomial in x.

iii. $\Phi_m(x,y) = \Phi_m(y,x)$ if $m > 1$.

iv. If m is a prime, l then, $\Phi_l(x,y) \equiv (x^l - y)(x - y^l) \bmod l\mathbb{Z}[x,y]$

Let l be the prime different from characteristic p, then the classical modular polynomial has $(l^2 + 3l + 4)/2$ coefficients. Here are examples of the classical modular polynomials taken from (Galin, 2007).

For l = 3, we have the modular polynomial as follows:

$$\begin{aligned}\Phi_3(x,y) = {} & x^4 - x^3y^3 + y^4 + 2232(x^3y^2 + x^2y^3) - 1069956(x^3y + xy^3) + 36864000(x^3 + y^3) \\ & + 2587918086x^2y^2 + 8900222976000(x^2y + xy^2) + 452984832000000(x^2 + y^2) \\ & - 770845966336000000xy + 1855425871872000000000(x + y)\end{aligned}$$

For l = 5, we have the following modular polynomial:

$$\begin{aligned}\Phi_5(x,y) = {} & x^6 - x^5y^5 + y^6 + 3720(x^5y^4 + x^4y^5) - 4550940(x^5y^3 + x^3y^5) + 2028551200(x^5y^2 + x^2y^5) \\ & - 246683410950(x^5y + xy^5) + 1963211489280(x^5 + y^5) + 1665999364600x^4y^4 \\ & + 107878928185336800(x^4y^3 + x^3y^4) + 38308360977981121537 5(x^4y^2 + x^2y^4) \\ & + 1285417989068288163840 00(x^4y + xy^4) + 1284733132841424456253440(x^4 + y^4) \\ & - 4412069655129148352461 00x^3y^3 + 2689848885838073157741772 8000(x^3y^2 + x^2y^3) \\ & - 1924579346189282829965510823116 8000(x^3y + xy^3) \\ & + 2802447778284395278043215652978 68800(x^3 + y^3) \\ & + 5110941777552418083110765199360000x^2y^2 \\ & + 3655473658394962929570647233265664 0000(x^2y + xy^2) \\ & + 6692500042627997708487149415015068467200(x^2 + y^2) \\ & - 2640734570766205962597157902479787829493 76xy \\ & + 5327433080342442545042016027335650915123 2000(x + y) \\ & + 1413599471547213586977534746910713627510046720 00.\end{aligned}$$

We now give some backgrounds needed for SEA algorithm. These information might have some gaps and readers are suggested to refer to (Silverman, 1986) and (Cox, 1989) for further details.

5.1.2.2 Elliptic curve over complex field

The theory of elliptic curves over complex field is corresponding to the lattice and thus equivalently to the torus that is the mapping of $E(\mathbb{C}) \to \mathbb{C}/\Lambda$ and $\mathbb{C}/\Lambda \to E(\mathbb{C})$. Lattice, $\Lambda = w_1\mathbb{Z} + w_2\mathbb{Z}$ where $w_1, w_2 \in \mathbb{C}$ are $\mathbb{R}$-linearly independent, then an elliptic function $f(z)$ defined on $\mathbb{C}$ except for isolated singularities, satisfies two conditions: $f(z)$ is meromorphic on $\mathbb{C}$ and $f(z + w_1) = f(z + w_2) = f(z)$. This indicates a doubly periodic meromorphic

function (Cox, 1989). An example of elliptic function is the Weierstrass $\wp$ – function defined in the following theorem. Proofs for all the theorems, lemma and propositions are omitted.

Theorem 5.1.2.2

Let $\Lambda \subset \mathbb{C}$ be a lattice. The Weierstrass $\wp$ – function relative to Λ is given by

$$\wp(z) = \wp(z, \Lambda) = \frac{1}{z^2} + \sum_{w \in \Lambda \setminus \{0\}} \left(\frac{1}{(z - w^2)} - \frac{1}{(w^2)} \right)$$

Then,

i. The sum defining $\wp(z)$ converges absolutely and uniformly on compact set not containing elements of Λ.
ii. $\wp(z)$ is meromorphic in $\mathbb{C}$ and has a double pole at each $w \in \Lambda$.
iii. $\wp(-z) = \wp(z)$, $\forall z \in \mathbb{C}$ which is an even function.
iv. $\wp(z + w) = \wp(z), \forall w \in \Lambda$

Therefore, the Weierstrass $\wp$ – function relative to Λ is a doubly periodic function with periods w_1 and w_2 which is known as the basis of Λ.

Theorem 5.1.2.3.

The relation between Weierstrass $\wp$ – function and its first derivative is given by $\wp'(z)^2 = 4\wp(z)^3 - g_2\wp(z) - g_3$. Then there is lattice, Λ such that $g_2(\Lambda) = 60G_4$ and $g_3(\Lambda) = 140G_6$ where $G_4 = \sum_{w \in \Lambda \setminus \{0\}} \frac{1}{w^4}$ and $G_6 = \sum_{w \in \Lambda \setminus \{0\}} \frac{1}{w^6}$. Hence, there is an isomorphism between points on elliptic curve over the complex field and points on the complex modulo a suitable lattice Λ that is $E(\mathbb{C}) \simeq \mathbb{C} / \Lambda$

5.1.2.3 j-invariant, $j(\tau)$

Elliptic function depends on the lattice being used. Let $\Lambda = w_1\mathbb{Z} + w_2\mathbb{Z}$ and $\tau = \frac{w_1}{w_2}$. Since $w_1, w_2 \in \mathbb{C}$ are

$\mathbb{R}$ -linearly independent, therefore the τ is not in $\mathbb{R}$. Now, τ belongs to Poincaré upper half plane,

$$\mathcal{H} = \{ x + iy \in \mathbb{C} \mid y > 0 \}.$$

By restricting to Λ_τ, we have $g_2(\Lambda) = g_2(\Lambda_\tau)$, $g_3(\Lambda) = g_3(\Lambda_\tau)$ and $D = g_2^3 - 27g_3^2$ which is closely related to the discriminant of the polynomial $4x^3 - g_2x - g_3$. Then,

$$j(\tau) = 1728 \frac{g_2(\tau)^3}{g_2(\tau)^3 - 27g_3(\tau)^2}.$$

If lattice $\Lambda \in \mathbb{C}$, there exists a nonzero $\lambda \in \mathbb{C}$ such that $\Lambda_\tau = \lambda\Lambda$ for some $\tau \in$ F where F is the standard fundamental region.

Two lattices, Λ and $\Lambda' \in \mathbb{C}$ is homothetic, then there exist nonzero $\lambda \in \mathbb{C}$ such that $\Lambda' = \lambda\Lambda$.

Theorem 5.1.2.4.

If Λ and Λ' are lattices in $\mathbb{C}$, then $j(\Lambda) = j(\Lambda')$ if and only if Λ and Λ' are homothetic.

$j(\tau)$ is a holomorphic function on the Poincaré upper half plane, $\mathcal{H} = \{x + iy \in \mathbb{C} \mid y > 0\}$. The properties of $j(\tau)$ are related to the action on the special linear group, $SL(2, \mathbb{Z})$ with determinant one on $\mathcal{H}$. This is defined as such that $z \in \mathcal{H}$ and $\gamma = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in SL(2,\mathbb{Z})$ then $\gamma\tau = \frac{a\tau + b}{c\tau + d}$, $\gamma\tau \in \mathcal{H}$. If τ and τ' in $\mathcal{H}$, then $j(\tau) = j(\tau')$ if and only if $\tau' = \gamma\tau$ for some $\gamma \in SL(2,\mathbb{Z})$

For classical modular polynomial and any $n > 0$.

$$S_n^* = \{\begin{pmatrix} a & b \\ 0 & d \end{pmatrix} \mid a,b,d \in \mathbb{Z}, 0 \le b < d, ad = n, \gcd(a,b,d) = 1\}$$

For $\alpha = \begin{pmatrix} a & b \\ 0 & d \end{pmatrix} \in S_n^*$, define the map

$$j \circ \alpha(\tau) = j(\frac{a\tau + b}{d})$$

Hence, the n-th modular polynomial can also defined as

$$\Phi_n(x, j) = \prod_{\alpha \in S_n^*} (x - j \circ \alpha(\tau)).$$

5.1.2.4 Computation of modular polynomial

Let l be a prime, we now discuss the method to compute modular polynomial, $\Phi_l(x,y)$. According to previous theorem, we have $\Phi_l(x,y) = \Phi_l(y,x)$ and $\Phi_l(x,y) \equiv (x^l - y)(x - y^l) \bmod l\mathbb{Z}[x,y]$. Besides, $\Phi_l(x,y)$ is a monic polynomial with degree l +1 as polynomial in x and therefore we can write

$$\Phi_l(x,y) = (x^l - y)(x - y^l) + l\sum_{0 \le i \le l} c_{ii} x^i y^i + l \sum_{0 \le i < j \le l} c_{ij}(x^i y^j + x^j y^i)$$

where the coefficient $c_{ij} \in \mathbb{Z}$ which can found by q-expansion of j –function. We also have the identity

$$\Phi_p(j(l\tau), j(\tau)) = 0 .$$

Substituting the *q*-expansion for $j(\tau)$ and $j(l\tau)$ into $\Phi_l(x,y)$, we have the following:

$$((j(l\tau)^l - j(\tau))(j(l\tau) - j(\tau)^l) + l\sum_{0\le i\le l} c_{ii} j(l\tau)^i j(\tau)^i + l\sum_{0\le i<j\le l} c_{ij}(j(l\tau)^i j(\tau)^j + j(l\tau)^j j(\tau)^i) = 0.$$

This is obtained by equating the coefficients of the different powers of infinite number of linear equations in the variable c_{ij}. However, the finite number of linear equations can be obtained by equating the coefficients of negative powers of q which is a unique solution. It is suffices to calculate those coefficients of the *q*-expansions which contribute to negative powers of $j(\tau)$ and only need the first $l^2 + l$ coefficients of the *q*-expansion of the *j*-function. Computing on modular polynomial becomes tedious as when prime l getting bigger, the number of digit for the coefficient do increase rapidly. Previously, we have listed the two modular polynomial $\Phi_3(x,y)$ and $\Phi_5(x,y)$. For $\Phi_{11}(x,y)$, its coefficients are more than 120 digits and is not shown here.

Lemma 5.1.2.1

Let $E_1/\mathbb{C}$ and $E_2/\mathbb{C}$ be two elliptic curves with *j*-invariants j_{E_1} and j_{E_2} respectively, then $\Phi_n(j_{E_1}, j_{E_2}) = 0$ if and only if there is an isogeny from E_1 to E_2 whose kernel is cyclic of degree n.

Theorem 5.1.2.5

Let E an elliptic curve defined over F_p with $p \neq l$, then the $l + 1$ zeroes $\tilde{j} \in F_p$ of the polynomial $\Phi_l(x, j(E)) = 0$ are the *j*-invariants of the isogenous curves $\tilde{E} = E/C$ with C one of the $l + 1$ cyclic subgroups of $E[l]$.

Theorem 5.1.2.6 (Atkin classification).

Let E be an ordinary elliptic curve defined over F_p with *j*-invariant $j \neq 0, 1728$. Let $\Phi_l(x, j) = h_1 h_2 ... h_s$ be the factorization of $\Phi_l(x, j) \in F_p[x]$ as a product of irreducible polynomials. Then there are the following possibilities for the degrees of $h_1, ..., h_s$:

i. $(1, l)$ or $(1, 1, \ldots, 1)$. In either case we have $\Delta = t^2 - 4p \equiv 0 (\mathrm{mod}\, l)$. In the former case we set $r = l$ and the later case $r = 1$.
ii. $(1, 1, r, r, ..., r)$. In this case $\Delta = t^2 - 4p$ is square modulo l, r divides $l - 1$ and ϕ_p acts on $E[l]$ as a diagonal matrix $\begin{pmatrix} \lambda & 0 \\ 0 & \mu \end{pmatrix}$ with $\lambda, \mu \in F_l^*$.
iii. $(r, r, \ldots, r)$ for some $r > 1$. In this case $\Delta = t^2 - 4p$ is a nonsquare modulo l, r divides $l + 1$ and the restriction of ϕ_p to $E[l]$ has an irreducible characteristic polynomial over F_l

In all these 3 cases, r is the order of ϕ_p in $\mathrm{PGL}_2(F_p)$ and the trace t satisfies $t^2 \equiv p(\xi + \xi^{-1})^2 (\mathrm{mod}\, l)$ for some r-th root of unity $\xi \in \overline{F}_l$. The number of irreducible factors s satisfies $(-1)^s = (\frac{p}{l})$.

Proof: Refer to (Galin, 2007).

To determine the type of prime, it suffices to compute $g(x) = \gcd(\Phi_l(x, j), x^p - x)$. If $g(x) \neq 1$, l is an Elkies prime else it is an Atkin prime.

Example 5.1.2.1

Let an elliptic curve, E defined over F_{113} with the equation given by $y^2 = x^3 - 15x + 13$

$S = \{2, 3, 5, 7\}$ such that $\prod_{i=1}^{4} l_i = 210 > \left\lceil 4\sqrt{p} \right\rceil = 43$. Let us check $l = 3$ and 5.

For $l = 3$. check whether it is an Elkies or Atkin prime. The j- invariant, $j_E = 28$.

$$\begin{aligned}\Phi_3(x,28) = {} & x^4 - x^3(28)^3 + (28)^4 + 2232(x^3(28)^2 + x^2(28)^3) - 1069956(x^3(28) + x(28)^3) \\ & +36864000(x^3 + (28)^3) + 2587918086x^2(28)^2 + 8900222976000(x^2(28) + x(28)^2) \\ & +452984832000000(x^2 + (28)^2) - 770845966336000000x(28) \\ & +1855425871872000000000(x + 28).\end{aligned}$$

$$\Phi_3(x,28) \equiv x^4 + 81x^3 + 111x^2 + 65x + 52 (\mathrm{mod}\, 113)$$

$$\gcd(\Phi_3(x,28), x^{113} - x) = 1$$

Hence 3 is an Atkin prime.

For $l = 5$. Check whether it is an Elkies or Atkin prime. The j- invariant, $j_E = 28$.

$$\Phi_5(x,28) \equiv x^6 + 90x^5 + 81x^4 + 65x^3 + 49x^2 \ (\mathrm{mod}\ 113)$$

$$\gcd(\Phi_5(x,28), x^{113} - x) = x^2 + 94x + 63 \neq 1$$

Hence 5 is an Elkies prime.

Definition 5.1.2.1

Let the discriminant of the characteristic equation, $\Delta = t^2 - 4p$. If Δ is a square in F_l then the prime l is an Elkies prime else l is an Atkin prime.

5.1.2.5 Atkin primes procedures

Since $t^2 = p(\xi + \xi^{-1})^2$ over F_l, each pair (ξ, ξ^{-1}) determines one value of t^2 or at most two values of t.

The number of the possible values of t_l is Euler totient function, $\varphi(r)$ and $r \leq l + 1$.

Let recall the reduced characteristic polynomial $\chi_l(T) = T^2 - t_l T + p_l = (T - \lambda)(T - \mu)$

$$(T - \lambda)(T - \mu) = T^2 - (\lambda\mu)T + (\lambda\mu)$$

Therefore $\lambda + \mu \equiv t(\mathrm{mod}\, l)$ and $\lambda\mu \equiv p(\mathrm{mod}\, l)$

Then $\frac{\lambda}{\mu}$ is an element of order exactly r in F_{l^2} Find r such that which $\gcd(\Phi_l(x), x^{p^r} - x) \neq 1$.

$\gamma = \frac{\lambda}{\mu}$ where $\gamma \in F_{l^2}$ is a primitive r-th root of unity. Now let g be a generator of $F_{l^2}^*$ and $\gamma_i = g^{\left(\frac{i(l^2-1)}{r}\right)}$ for $\gcd(i, r) = 1$ and satisfying $1 \leq i < r$.

Next, for nonsquare $d \in F_l$, we have $\lambda = x_1 + x_2\sqrt{d}$ and $\mu = x_1 - x_2\sqrt{d}$ for some $x_i \in F_l$. Similarly we have $\gamma_i = g_{i_1} + g_{i_2}(\sqrt{d}) = \frac{\lambda}{\mu} = \frac{\lambda^2}{\lambda\mu} = \frac{x_1^2 + x_2^2 d + 2x_1x_2\sqrt{d}}{p}$ where $g_{i_1}, g_{i_2} \in F_l$

Compare both sides,

$$g_{i_1} \equiv \frac{x_1x_2 + x_2^2 d}{p}(\mathrm{mod}\, l) \qquad \Rightarrow pg_{i_1} \equiv x_1^2 + x_2^2 d(\mathrm{mod}\, l)$$

$$g_{i_2}\sqrt{d} \equiv \frac{2x_1x_2\sqrt{d}}{p}(\mathrm{mod}\, l) \;\Rightarrow pg_{i_2} \equiv 2x_1x_2(\mathrm{mod}\, l)$$

Also, $p \equiv \lambda\mu \equiv x_1^2 + dx_2^2(\mathrm{mod}\, l)$, so it follows that $x_1^2 = \frac{p(g_{i_1} + 1)}{2}$. If x_1^2 is not a square in F_l, γ_i is discarded and move to the next one. Else, we have the following.

$$\therefore t \equiv \lambda + \mu \equiv 2x_1(\mathrm{mod}\, l)$$

5.1.2.6 Elkies primes procedures

Determine for the isogenous elliptic curve. Then recall the reduced characteristic polynomial $\chi_l(T) = T^2 - t_lT + p_l = (T - \lambda)(T - \mu)$

$$(T - \lambda)(T - \mu) = T^2 - (\lambda\mu)T + (\lambda\mu).$$

Therefore, $\lambda + \mu \equiv t(\mathrm{mod}\, l)$ and $\lambda\mu \equiv p(\mathrm{mod}\, l)$.

Notice that $t_l \equiv \lambda + \mu = \lambda + {}^{p}\!/_{\lambda}\,(\mathrm{mod}\, l)$, so once we get the value of λ then we can find t_l.

If $\lambda = \mu$, then $t_l \equiv 2\lambda \equiv 2\sqrt{p}(\mathrm{mod}\, l)$.

If $\lambda \neq \mu$, $E[l]$ has two subgroups C_1, C_2 that are stable under ϕ_p, we need to replace the division polynomial with degree $(l^2 - 1)/2$ by finding a kernel polynomial with degree $(l - 1)/2$ whose roots are the x-coordinate of the subgroup C_1 or C_2. The kernel polynomial is defined by

$$F_l(x) = \prod_{\pm P \in C_L \setminus \{0\}} (x - x(P))$$

One can obtain the value of λ by using the relation $(x^p, y^p) = \lambda(x, y)$.

We find λ such that $\gcd(\psi_\lambda^2(x^p - x) + \psi_{\lambda-1}\psi_{\lambda+1}, F_l(x)) \neq 1$. It suffices to check the value $1 \leq \lambda \leq (l-1)/2$. Then compute $t_l \equiv \lambda + \mu = \lambda + p/\lambda \pmod{l}$.

To combine Elkies and Atkin prime, we apply the concept of Chinese Remainder Theorem, and then we obtain the *t* in the final step by using the elementary method (baby-steps giant-steps which is not covered in this chapter, and one should refer to (Galin, 2007) for further details). Next, we summarize these two algorithms into making some comparison between them.

6. Compare and contrast between Schoof's algorithm and Schoof-Elkies-Atkin (SEA) algorithm

i. Similarities between Schoof's algorithm and SEA algorithm

- Polynomial time and deterministic point counting algorithms for elliptic curve with characteristic $K \neq 2, 3$
- Using Hasse's theorem or specifically Hasse's interval as a boundary to determine $\#E(F_p)$.
- Using a specific type of polynomial which is essential for the computation steps.
- Classified as *l*-adic point counting algorithms.
- Begin by letting $S = \{2, 3, 5, \ldots, L\}$ be a set of primes not including char (F_p) such that $\prod_{l \in S} l > 4\sqrt{p}$.
- Both algorithms is to find the trace of the Frobenius endomorphism, *t.*
- Making use of the Frobenius endomorphism, ϕ_p and the characteristic equation is such that:

$$\chi_l(T) = T^2 - t_l T + p_l = (T - \lambda)(T - \mu), \text{ so } t \equiv \lambda + \mu \pmod{l} \text{ and } p \equiv \lambda\mu \pmod{l}.$$

ii. Differences between Schoof's algorithm and SEA algorithm

- Practicality

i. Schoof's algorithm: Most successful general point counting algorithm but not practical because the degree of division polynomial will grow exponentially when *l* becomes larger.

ii. SEA algorithm: Most practical version of point counting algorithm. However the use of classical modular polynomial will lead to the increasing of number of coefficients when *l* becomes larger. This problem is overcome by using canonical modular polynomial, Müller modular polynomial and Atkin modular polynomial which have similar construction like classical modular polynomial.

- Complexity

i. Schoof's algorithm*: O* $(\log^8 p)$ bit operations.

ii. SEA algorithm: O (log $^6 p$) bit operations due to the replacement of division polynomial $(l^2-1)/2$ by its factor that is kernel polynomial with degree $(l-1)/2$.

- Classification of prime, p

i. Schoof's algorithm: No classification of prime. However, two cases are considered such that :

$$(x^{p^2}, y^{p^2}) \neq \pm p_l(x,y) \text{ or } (x^{p^2}, y^{p^2}) = \pm p_l(x,y)$$

ii. SEA algorithm: for $p > 2$, p is classified as Elkies primes or Atkin primes by using modular polynomial such that $\gcd(\Phi_l(x, j(E)), x^p - x) = 1$, then l is an Atkin prime, else l is an Elkies prime.

- Polynomial involved

i. Schoof's algorithm: division polynomial, ψ_l with degree $(l^2-1)/2$. To construct division polynomial, concept of torsion point is applicable.

ii. SEA algorithm: modular polynomial with degree l +1 is used to differentiate Atkin and Elkies prime. The construction of modular polynomial works in complex field and also need to deal with j-function and q-expansion (Cox, 1989) but the result can be applied in finite field, F_p.

- Method to combine the t_l (mod l)

i. Schoof's algorithm: Recover the t from t_l (mod l) from Chinese Remainder Theorem.

ii. SEA algorithm:

For Elkies primes: Recover the t_E from t_l (mod l) from Chinese Remainder Theorem. For Atkin primes: Divide the primes into two sets that each in equal numbers by using Chinese Remainder Theorem. Finally this theorem is used again and then the exact t is found by using baby-steps giant steps.

7. Some literature on Schoof and Schoof-Elkies-Atkin (SEA) point counting algorithms

In this section, we will give some brief literature of these two algorithms. As we have mentioned earlier, René Schoof (Schoof, 1985) had proposed the Schoof's algorithm in 1985. In (Cohen et al, 2006), Atkin and Elkies had further improved the Schoof's algorithm in 1991. In Elkies procedure, Elkies had replaced the division polynomial with degree $(l^2-1)/2$ by a kernel polynomial with degree $(l-1)/2$ whereas Atkin developed an algorithm to evaluate the order of ϕ_p in $PGL_2(F_l)$ and hence shown that the number of point can be counted on $E(F_p)$ and this thus lead to the Schoof-Elkies-Atkin (SEA) algorithm which was practical.

In (Menezes et al., 1993), elliptic curves which defined over field of characteristic 2 are attractive because the arithmetic easier for implementation. They have employed some heuristic to improve the running time and able to compute $\#E(F_{2^m})$ for $m \leq 155$. For the Schoof's part, they were able to compute t modulo l for l = 3, 5, 7, 11, 13, 17, 19, 23, 31, 64, 128, 256, 512 and 1024. The computation on $\#E(F_{2^{155}})$ takes roughly 61 hours on a SUN-2

SPARC station. Then, it was also mentioned that the information obtained from Schoof's algorithm and the heuristics can be combined with the information from Atkin's method to compute $\#E(F_{2^m})$ for large values of m.

In (Couveignes & Morain, 1994), they had shown how to use the powers of good prime in an efficient way by computing the isogenies between curves over the ground field. They had investigated the properties of new structure which is known as isogeny cycle.

In (Lercier & Morain, 1995), they mentioned that when l was an Elkies prime, the cost of computation turn out to be greater than that computation of Atkin prime and hence suggested that it is better to treat an Elkies prime as an Atkin prime and hence motivate their dynamic strategy. The implementation result shown that Schoof's algorithm in characteristic 2 was faster than in large characteristic at least for small fields. The large prime case was faster due to the polynomial arithmetic was faster for F_{2^n} since squaring was an easy operation in characteristic 2. However, when n increased, the computing cost of the isogeny took much time than in large prime case.

In (Lercier, 1997), mentioned the improvement made by Elkies and Atkin and worked in any finite field. The computation of isogeny is only worked in finite fields of large characteristic. However this problem was solved by Couveignes, by taking in the formal group and had implemented it. The computation of isogenies then turned out to be the major cost while counting the point. Lercier had proposed better algorithm for characteristic 2 case which based on algebraic properties. The slight change in Schoof's algorithm sped up the randomly search of elliptic curves with order nearly prime instead of specific curves such as supersingular curves or curves obtained from complex multiplication.

In (Izu et al., 1998), they wanted to find elliptic curve which had prime order and believed that curve with this order was secure for cryptographic application. In calculating the order, they combined efficiently the Atkin and Elkies method, the isogeny cycles method and trial search by match-and-sort techniques and implemented them for elliptic curve over prime field, F_p in a reasonable time where p is a prime number whose size around 240-bits. As a result, it had increased the speed of the process almost 20%. They managed to find elliptic curves with prime order in a reasonable time for characteristic p of base field is around 240- bits.

In SEA algorithm, the classical modular polynomials with degree l +1 will increase the size of coefficient as l increases, as well as their degree in y also is very high. Therefore canonical modular polynomials achieve small coefficient and lower degree in y. Details can be obtained in (Cohen et al., 2006). Besides, according to the work from (Blake et al., n.d.), their approach shown that classical modular polynomial can be replaced by Müller modular polynomial or Atkin modular polynomial. This experiment had been done for l =197. The result shows that Müller modular polynomial has less number of coefficients compared to the classical one. However the Atkin modular polynomial has the least number of coefficients compared with the classical modular polynomial and Müller modular polynomial.

8. Conclusion

This chapter gives some backgrounds on elliptic curve cryptography. The mathematical preliminaries on elliptic curve, basic definitions, group operations on an elliptic curve, the

addition law as well as the doubling operations are part of the discussion topics in this chapter. This chapter also includes the arithmetic of elliptic curves defined over the real numbers as well as on a finite field and some examples are shown to enhance understanding. Several schemes such as elliptic curve Diffie-Hellman key exchange scheme, elliptic curve ElGamal cryptosystem and elliptic curve digital signature scheme are discussed along with some examples. Concept of point counting algorithms is also treated quite rigorously in terms of the mathematical aspects, and the discussion is restricted to two types of algorithms, the Schoof and the Schoof-Elkies-Atkin (SEA) point counting algorithms. Building on the discussion of the point counting algorithms, several comparisons are derived along with some literatures on the development of these two point counting algorithms especially on the Schoof-Elkies-Atkin (SEA) algorithm. This chapter has shown the procedures in the Schoof and the Schoof-Elkies-Atkin (SEA) algorithms. Extensive mathematical concepts explaining these two algorithms are displayed in this chapter. The Schoof point counting algorithm is regarded as an initiative effort towards producing efficient point counting algorithm, where several modification has emerges from the idea of this algorithm, and the immediate improvement were produced by Elkies and Atkin. The most recent known modification build on Schoof algorithm is the one from Pierrick Gaudry, David Kohel, Benjamin Smith (Schoof-Pila algorithm), presented in the Elliptic Curve Cryptography workshop, held in Nancy, France, in September 2011.

The arithmetic on elliptic curve plays a very important role in cryptography and this chapter has highlighted some mathematical aspects needed in the development of elliptic curve cryptography. Many studies have been devoted to finding fast algorithms on performing group operations on elliptic curves as well as algorithms to compute number of points on elliptic curves. So far elliptic curve cryptography seems to out perform other cryptographic schemes. Interest groups working on elliptic curve cryptography are seen to have more ideas to explore as most directions are on the higher genus curves or hyperelliptic curves instead of the ordinary curves that are being treated in this chapter. Genus 2 curve for instance, is a hyperelliptic curve, which possesses different properties from the ordinary elliptic curve. Points on hyperelliptic curves do not forms a group, instead the corresponding jacobian takes the role. Some properties in the ordinary curves could be extended to those higher genus curves. In the future, we might probably have a situation where hyperelliptic curve cryptography comes into play.

9. Acknowledgment

This article was written under the funding of the Universiti Sains Malaysia Short Term Grant, 2010-2012, account number 304/PMaths/6310075.

10. References

Blake, I. F., Csirik, J. A., Rubinstein, M., & Seroussi, (n.d.), G. On the Computation of Modular Polynomials for Elliptic Curves. *HP Laboratories Technical Report.*

Chen, R. J. (2008). Lecture Notes on Elliptic Curve Crytography. Department of Computer Science, National Chiao Tung University.

Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Kim, N., et al. (2006). *Handbook of Elliptic Curve and Hyperelliptic Curve Cryptography*, Taylor & Francis Group, LLC., ISBN:1-58488-518-1 , New York.

Couveignes, J., & Morain, F. (1994). Schoof's Algorithm and Isogeny Cycles. In: *Algorithmic Number Theory*, L. M. Adleman & M. D. Huang, pp. 43-58, Springer Berlin/Heidelberg, ISBN:978-3-540-58691-3, New York.

Cox, D. A. (1989). *Primes of the Form $x^2 + ny^2$: Fermat, Class Field Theory and Complex Multiplication*, John Wiley & Sons, Inc., ISBN: 0-471-50654-0, USA.

Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on information Theory,* Vol 22, No 6, (November 1976), pp. 644-654, ISSN: 0018-9448.

Galin, B. (2007). *Schoof-Elkies-Atkin Algorithm,* Senior thesis, Department of Mathematics, Stanford University, USA.

Izu, T., Kogure, J., Noro, M., & Yokoyama, K. (1998). Efficient Implementation of Schoof's Algorithm, In: *Advances in Cryptology — ASIACRYPT'98 International Conference on the Theory and Application of Cryptology and Information Security*, Kazuo Ohta & Dingyi Pei, pp. 66-79, Springer Berlin / Heidelberg, ISBN: 978-3-540-65109-3, New York.

Jacobson, M. J., Jr., Erickson S., Hammer, J. , Scheidler, R., Shang, N., Shen, S. & Stein, A. (2009). Cryptographic Aspects of Real Hyperelliptic Curves, In: *13th Elliptic Curves Cryptosystem Workshop*, Calgary, Canada.

Koblitz, N. (1987) Elliptic Curve Cryptosystems. Mathematics of Computation, Vol. 48, No. 177, (January 1987), pp. 203-209.

Lawrence, C. W. & Wade, T.(2006). *Introduction to Cryptography with Coding Theory, 2nd edition,* Pearson Prentice Hall, ISBN: 0-13-186239-1, USA.

Lercier, R. (1997). Finding Good Random Elliptic Curves for Cryptosystems Defined over F_{2n}, *EUROCRYPT '97 International Conference on the Theory and Application of Cryptographic Techniques* , ISBN: 3-540-62975-0, Konstanz, Germany, May 11–15, 1997.

Lercier, R., & Morain, F. (1995). Counting the Number of Points on Elliptic Curves over Finite Fields: Strategies and Performances, *EUROCRYPT'95 Proceedings of the 14th Annual International Conference on Theory and Application of Cryptographic Techniques,* ISBN: 3-540-59409-4, Saint-Malo, France, May 21-25, 1995.

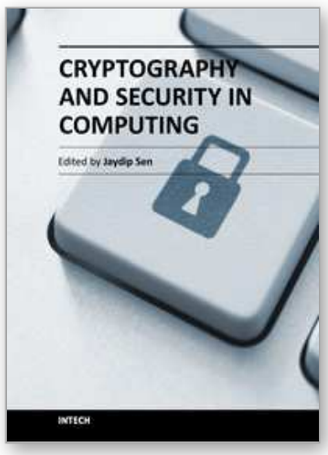
McGee, J. J. (2006). *René Schoof's Algorithm for Determining the Order of the Group of Points on an Elliptic Curve over a Finite Field.* Virginia Polytechnic Institute and State University, USA.

Miller, V. (1986). Use of Elliptic Curves in Cryptography, Advances in Cryptology — CRYPTO '85 Proceedings, Hugh C. Williams, pp. 417-426, Springer Berlin / Heidelberg, ISBN: 978-3-540-16463-0, New York.

Menezes, A., Vanstone, S., & Zuccherato, R. (1993). Counting Points on Elliptic Curves over F_{2m}. *Mathematics of Computation,* Vol. 60, No. 201, (January 1993), pp. 407-420, DOI 10.1090/S0025-5718-1993-1153167-9.

Schoof, R. (1985). Elliptic Curves over Finite Fields and the Computation of Square Roots Mod *p*, *Mathematics of Computation, Volume* 44, No. 170, (April 1985), pp. 483-494.

Silverman, J. H. (1986). *Graduate Texts in Mathematics*: *The Arithmetic of Elliptic Curves,* Springer-Verlag, ISBN: 0-387-96203-4, USA.



Cryptography and Security in Computing

Edited by Dr. Jaydip Sen

ISBN 978-953-51-0179-6
Hard cover, 242 pages
Publisher InTech
Published online 07, March, 2012
Published in print edition March, 2012

The purpose of this book is to present some of the critical security challenges in today's computing world and to discuss mechanisms for defending against those attacks by using classical and modern approaches of cryptography and other defence mechanisms. It contains eleven chapters which are divided into two parts. The chapters in Part 1 of the book mostly deal with theoretical and fundamental aspects of cryptography. The chapters in Part 2, on the other hand, discuss various applications of cryptographic protocols and techniques in designing computing and network security solutions. The book will be useful for researchers, engineers, graduate and doctoral students working in cryptography and security related areas. It will also be useful for faculty members of graduate schools and universities.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen