Elliptic Curve Cryptography

Dissertation

Integrated Masters of Science
In
Applied Mathematics

Submitted by
Harsh Kumar Chourasia

supervised by
Dr. Ram Krishna Pandey

Department of Mathematics
Indian Institute of Technology, Roorkee
Department of Mathematics
Roorkee-247667

Declaration

I hereby certify that the work which is being presented in the thesis entitled **Elliptic Curve Cryptography** in the partial fulfillment of the requirement for the award of the degree of Integrated Master of Science in Applied Mathematics and submitted to the Department of Mathematics, Indian Institute of Technology Roorkee is an authentic record of my work carried out during a period from January 2022 to April 2022 under the supervision of **Dr. R.K. Pandey**, Associate Professor, Mathematics Department, Indian Institute of Technology Roorkee. The matter presented in this report has not been submitted by me for the award of any other degree of this or any other institute.

Harsh Kumar Chourasia
I. M.Sc, Applied Mathematics
Department of Mathematics
IIT Roorkee
Date:

CERTIFICATE

This is certified that the above statement made by the candidate is correct to the best of my knowledge.

Dr. RK Pandey
Associate Professor
Department of Mathematics
IIT Roorkee
Date:

Abstract

This text discusses Cryptography using Elliptic Curves. It has many practical applications in end-to-end encryption, data and password storing, and blockchain technology. This dissertation aims to understand how the elliptic curve is used for cryptography. This dissertation is divided into five chapters. The first chapter discusses the fundamental concepts from abstract algebra and number theory that are required to study elliptic curve cryptography. The second chapter provides an introduction to cryptography and elliptic curves. The third chapter deals with the Elliptic Diffie-Hellman key exchange. The fourth chapter discusses the implementation of Elliptic Curves using javascript. Finally, the fifth chapter is used to conclude the report.

Acknowledgments

I would like to thank my supervisor, Ram Krishna Pandey, Department of Mathematics, Indian Institute of Technology Roorkee, for allowing me to work under his guidance. His scholarly advice, meticulous scrutiny, and encouragement have helped me greatly accomplish this task.

I would finally like to thank Prof Premananda Bera, Head of Department, Department of Mathematics, Indian Institute of Technology, for permitting me to carry out this work at IIT Roorkee.

Harsh Kumar Chourasia
I. M.Sc, Applied Mathematics
Department of Mathematics
IIT Roorkee
Date:

Contents

Declaration i
Abstract ii
Acknowledgments iii

1 Fundamental concepts **2**
1.1 Group 2
1.1.1 Abelian Group 3
1.2 Ring 3
1.3 Field 3
1.4 Fermat's little theorem 4

2 Elliptic Curves and Cryptography **6**
2.1 Introduction to Cryptography 6
2.1.1 Symmetric cryptography 7
2.1.2 Asymmetric cryptography 8
2.2 Elliptic Curves 9
2.3 Elliptic curves over finite fields 13
2.4 Elliptic curve discrete logarithm problem 14
2.4.1 Double and Add Algorithm 15
2.4.2 Hasse's Theorem 17
2.4.3 How to solve ECDLP 17
2.5 Elliptic Diffie–Hellman key exchange 18
2.6 ElGamal Elliptic Curve Cryptosystem 19

3 Implementation **21**

Chapter 1

Fundamental concepts

For understanding Elliptic Curve Cryptography basic understanding of Abstract Algebra, Number Theory, and probability is required. This chapter discusses the definition of algebraic structures such as group, ring, and field and the statement and proof of the famous Fermat's Little Theorem.

1.1 Group

Definition 1. *The set G equipped with single operation $*$ such the four properties given below are satisfied is called a Group.*
(1) *Closure:* $\forall x, y \in G, x * y \in G$
(2) *Additive identity:* $\exists 0 \in G$, *such that* $\forall x \in G$, $0 * x = x * 0 = x$
(3) *Associative Property:* $\forall x, y, z \in G$, $(x * y) * z = x * (y * z)$
(4) *Inverse:* $\forall x \in S, \exists y \in G$ *such that* $x * y = 0$ *where* y *is known as inverse of* x *and is denoted by* x^{-1}

1.1.1 Abelian Group

Definition 2. *The set G is equipped with single operation $*$ such the properties given below are satisfied is called a Abelian Group.*
(1) *Closure:* $\forall x, y \in G, x * y \in G$
(2) *Additive identity:* $\exists 0 \in G$, *such that* $\forall x \in G$, $0*x = x*0 = x$
(3) *Associative Property:* $\forall x, y, z \in G$, $(x * y) * z = x * (y * z)$
(4) *Commutative Property:* $\forall x, y \in G$, $x * y = y * x$
(5) *Inverse:* $\forall x \in G, \exists y \in G$ *such that* $x * y = 0$ *where* y *is known as inverse of* x *and is denoted by* x^{-1}

So, a abelian group G is a group with $\forall x, y \in G, x * y = y * x$

1.2 Ring

Definition 3. *Ring is a set R with two operations $+$ and $*$ which satisfy the below properties*
(1) *It is abelian group under* $+$
(2) *Closure under* $*$: $x, y \in R \Rightarrow x * y \in R$
(3) *Associative under* $*$: $x, y, z \in R \Rightarrow (x * y) * z = x * (y * z)$
(4) *Distributive property* $x, y, z \in R$

$$x * (y + z) = x * y + x * z$$
$$(x + y) * z = x * z + y * z$$

1.3 Field

Definition 4. *Field is a set F with two operations $+$ and $*$ with following properties*
(1) *Commutative group under* $+$
(2) *Commutative group under* $*$
(3) *Distributive property* $x, y, z \in F$

$$x * (y + z) = x * y + x * z$$
$$(x + y) * z = x * z + y * z$$

1.4 Fermat's little theorem

Theorem 1. *Let p be any prime number. For any number a such that $p \nmid a$. Then $a^{p-1} \equiv 1 \pmod{p}$*

Proof. Assume p is a prime number and $p \nmid a$
Every integer is congruent to one of $0, 1, 2, \cdots, p-1 \pmod{p}$
Only focus on non zero congruence classes, because $0 \pmod{p}$ contains all the multiples of p (and $p \nmid a$). Focus on $0, 1, \cdots, p-1$
Multiply all of these by a:

$$a, 2a, \cdots, (p-1)a$$

Show that this is a rearrangement of $0, 1, 2, \cdots, p-1$
Case 1: None of these are congruent to 0.
Suppose $r.a \equiv 0 \pmod{p}$
Then $p \nmid r.a$, but this is impossible since $p \nmid a$ and $r < p$
Case 2: These are distinct, no two are congruent to each other.
Pick two values $r.a$,$s.a$

$$0 < r < p$$

$$0 < s < p$$

Let's show that $r.a \not\equiv s.a \pmod{p}$
So look at $r.a - s.a = (r-s).a$. As $p \nmid a$, so can $p \mid (r-s)$?

$$0 < r < p$$

$$-p < -s < 0$$

Adding these inequalities gives you:

$$-p < r - s < p$$

So, $p \nmid (r-s)$ which means $a, 2a, \cdots, (p-1)a$ is a rearrangement of

$1, 2, \cdots, (p-1)$.

$$a, 2a, \cdots, (p-1)a \equiv 1, 2, \cdots, (p-1) \pmod{p}$$

$$(p-1)!a^{p-1} \equiv (p-1)! \pmod{p}$$

$$a^{p-1} \equiv 1 \pmod{p}$$

□

Chapter 2

Elliptic Curves and Cryptography

2.1 Introduction to Cryptography

The written word is the most important invention in human history. However, as long as humans can share information, they also need to hide information. This has led to the invention of cryptography.
The word cryptography comes from Greek which means "hidden writing". According to Wikipedia [3], **Cryptography, or cryptology, is the practice and study of techniques for secure communication in the presence of adversarial behavior.**
Some of the application of Cryptography includes:

- **End-to-end Encryption** for e-mail, messaging apps, GSM phones.
- **Storing Data**: One of the most significant consumer applications of cryptography includes Kindle, and iPod, which store books and songs in an encrypted format to protect copyright.
- **Storing Password**: Storing passwords in plain text is not

secure. If an attacker gains access to the system, they can read the password. If the password is converted into a hash using a one-way mapping function and stored. Every time a user logs in, the password will be converted into the hash and compared with the stored password.

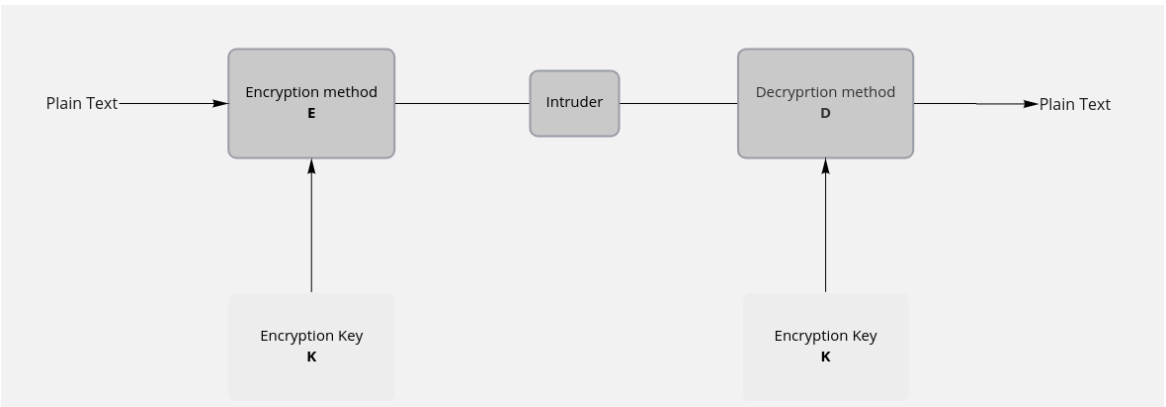
There are main types of cryptography: symmetric key cryptography and asymmetric key cryptography.

2.1.1 Symmetric cryptography

Let Alice want to share a message with Bob. They do so by using a shared key and knowledge of some algorithm to encrypt and decrypt the message. Alice encrypts the message using the key to produce the ciphertext. Now Bob can use the key with the ciphertext to decrypt the message.

In symmetric cryptography, a shared key is used by the sender and receiver.

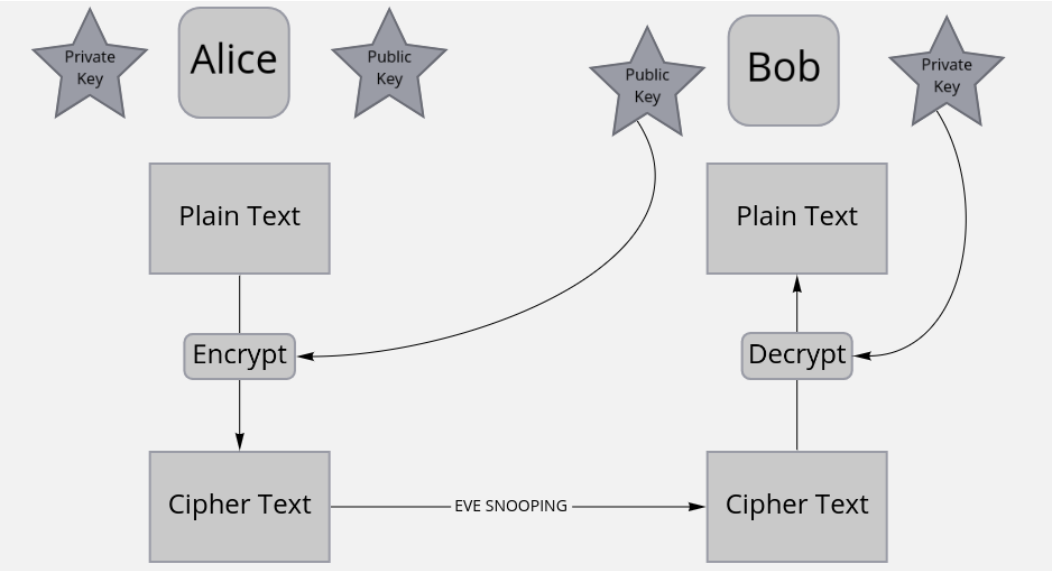
Figure 2.1: A message exchange using a common key K



2.1.2 Asymmetric cryptography

Asymmetric cryptography works by using private and public key pairs. Each user has a private, public key pair. The public key can be shared freely across the network and is used to verify the owner of a message. The private key is not transmitted across the network. Public key are used to encrypt the message and the private key is used to decrypt the message. The major advantage of asymmetric cryptography is that there is no need for a shared key.

Figure 2.2: A message exchange using private and public keys



2.2 Elliptic Curves

Equation of type $y^2 = x^3+ax+b$ are called Weierstrass equation. It is named after Karl Weierstrass $(1815 - 1897)$ who studied them in 19^{th} century.

Definition 5. *Elliptic curves are solution sets of Weierstrass equations*

$$E : y^2 = x^3 + ax + b...(1)$$

with $\{\ \mathcal{O}\ \}$ *where* $\Delta_E = 4a^3 + 27b^2 \neq 0$. $\Delta_E \neq 0$ *guarantees that the equation* $x^3 + ax + b$ *has no repeated roots i.e.* $x^3+ax+b = (x-e_1)(x-e_2)(x-e_3)$ *where* e_1, e_2, e_3 *are distinct.* $\mathcal{O}$ *is defined as the point at infinity which lies on every vertical line.*

Figure 2.3: Example of Elliptic Curves

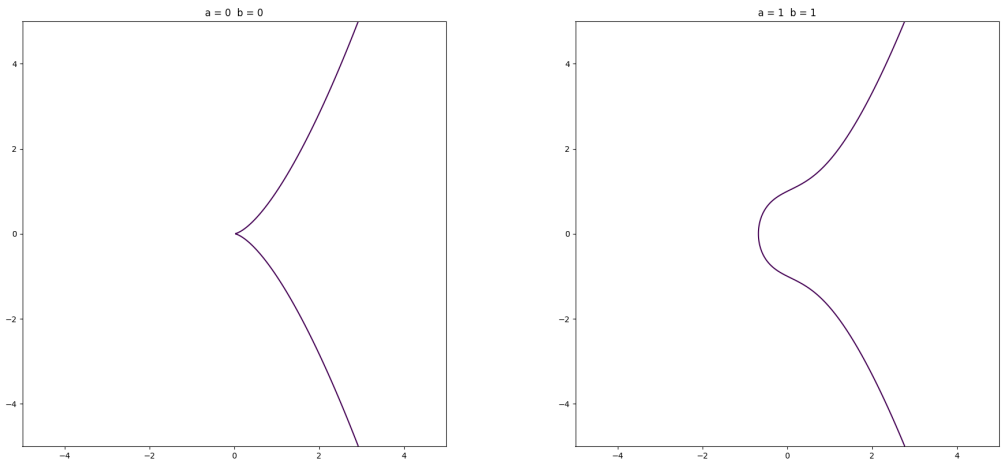
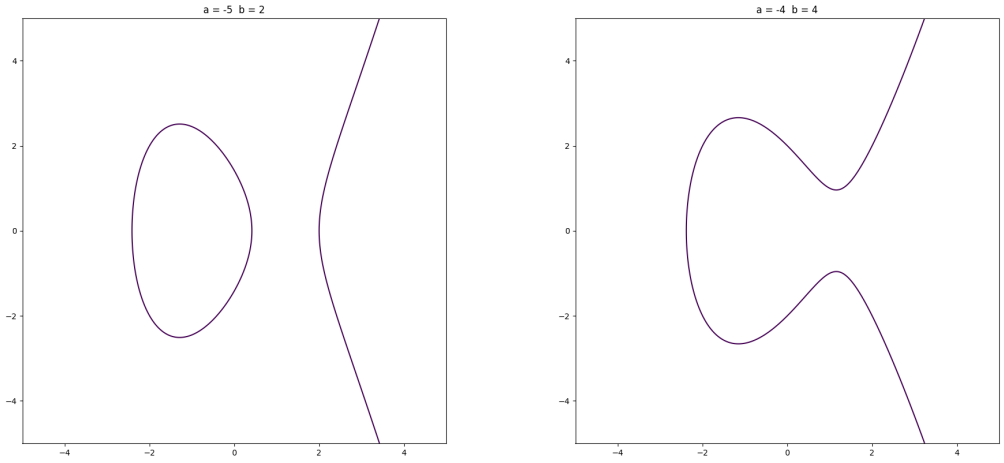


Figure 2.4: Example of Elliptic Curves



If (x,y) satisfies eq(1), then (x,-y) is also a solution of equation (1). So, elliptic curves are symmetric about x-axis.
The definition of addition "+" operator is a not the usual definition one might expect

$$(a, b) + (c, d) \neq (a + c, b + d)$$

Let P_1 and P_2 are two points on elliptic curve. If we make a line L that passes through P_1 and P_2, it will intersect the curve at point $P_3 = (x_3, y_3)$. The reflection of P_3 from x-axis i.e. $(x_3, -y_3)$ is called the sum of points P_1 and P_2.

So, what is $P_1 + P_1$? This is the limiting case where $P_2 \to P_1$. Line L becomes the tangent to E at P_1. This line will intersect E at P_3. The reflection of P_3 about x-axis is $P_1 + P_1$.

Let's look at the case when two points on the curve when $P_1 = (x, y)$ and $P_2 = (x, -y)$ are added. In this case line L is $x = a$. L will not intersect the curve at third point. In this case we define

Figure 2.5: Example of $P_1 + P_2$

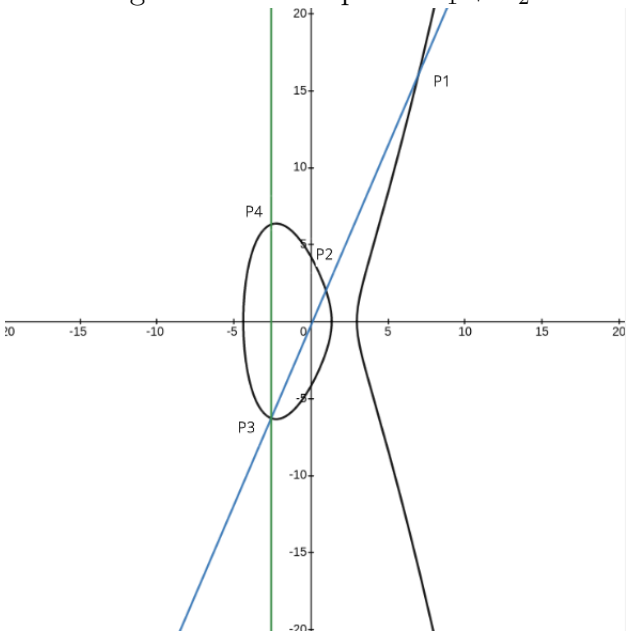
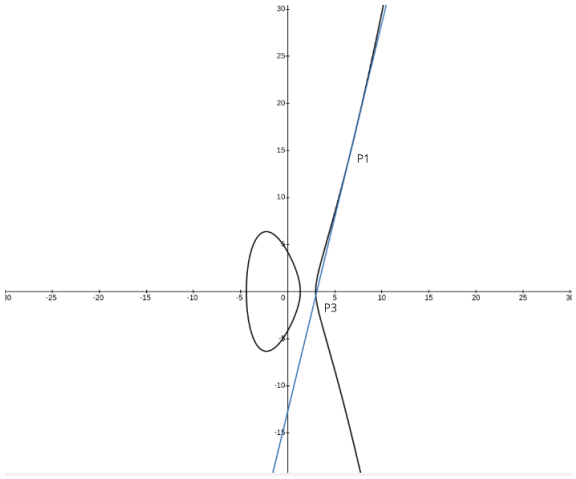
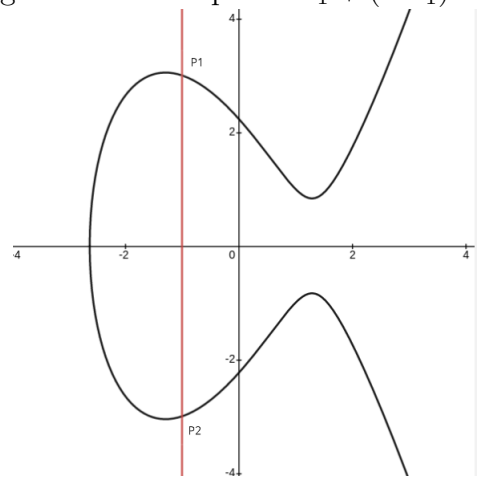


Figure 2.6: Example of $P_1 + P_1$



$P_1 + P_2 = \mathscr{O}$. We define $\mathscr{O}$ as the point in infinity that lies on every vertical line. If P = (x,y) then -P is defined as (x,-y). So, $P + (-P) = \mathscr{O}$...(2)

Figure 2.7: Example of $P_1 + (-P_1) = \mathscr{O}$



Theorem 2. *Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be two points on elliptic curve $E : Y^2 = X^3 + AX + B$. Then the following are true:*

1. *If $P = \mathscr{O}$, then $P + Q = Q$*
2. *If $Q = \mathscr{O}$, then $P + Q = P$*
3. *If $P = -Q$ then $P + Q = \mathscr{O}$*
4. *If $P \neq Q$ then $\lambda = (y_2 - y_1)/(x_2 - x_1)$ and if $P = Q$ then $\lambda = (3x_1^2 + A)/(2y_1)$. In both cases,*
   $P_1 + P_2 = (\lambda^2 - x_1 - x_2, \lambda(x_1 - x_3) - y_1)$

Proof. (1), (2), (3) are true as discussed above.
(4) If $P \neq Q$ then λ is the slope of the line passing through P and Q. If $P = Q$ then λ is the slope of the tangent at P. Suppose Line $y = \lambda x + c$ intersects the curve at (x_3, y_3) in addition to (x_1, y_1) and (x_2, y_2).

$$(\lambda x + c)^2 = x^3 + Ax + B$$

$$x^3 - \lambda^2 x^2 + (A - 2c\lambda)x + (B - c^2) = (x - x_1)(x - x_2)(x - x_3)$$

We get $x_3 = \lambda^2 - x_1 - x_3$ by comparing the coefficients of x^2. $y_3 = \lambda x + c = y_1 - \lambda(x_1 - x_2)$. The ordinate of $P + Q$ is $-y_3 = \lambda(x_1 - x_3) - y_1)$ □

Theorem 3. *Let E be Elliptic curve. Then E forms abelian group under addition. The following statements are true:*

1. $P_1 + \mathscr{O} = \mathscr{O} + P_1 = P_1$ *for all* $P_1 \in E$
2. $P_1 + (-P_1) = \mathscr{O}$ *for all* $P_1 \in E$
3. $(P_1 + P_2) + P_3 = P_1 + (P_2 + P_3)$ *for all* $P_1, P_2, P_3 \in E$
4. $P_1 + P_2 = P_2 + P_1$ *for all* $P_1, P_2 \in E$

Proof. (1) Claim: $P_1 + \mathscr{O} = P_1$
If a line is drawn through P and $\mathscr{O}$, it will intersect E at -P. Reflection of -P from x-axis is again P. So, $P_1 + \mathscr{O} = P_1$ Similarly, $\mathscr{O} + P_1 = P_1$
(2) Explained above in equation (2)
(3) Associative property is non-trivial to prove using geometry. It can be verified by using Theorem 2 by calculation using substitution.
(4) is true as line passing through P_1 and P_2 is same as the line passing through P_2 and P_1. □

2.3 Elliptic curves over finite fields

Definition 6. *Elliptic curve over a finite field F_p is defined as equation of the form*

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

where $a, b \in F_p$ and $\Delta_E = 4a^3 + 27b^2 \neq 0$.

Example: Let's suppose $a = 0$, $b = 1$ and $p = 11$.
$F_{11} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}.\{a, b\} \in F_{11}$ and $4 * 0^2 + 27 *$

$1^3 = 27 \neq 0$. Hence, $E : y^2 = x^3 + 1 \pmod{11}$ is an elliptic curve over finite field.
$E(F_{11}) = \{\mathscr{O}, (10,0), (0,10), (0,1), (9,2), (9,9), (2,3), (2,8), (6,8), (8,4), (8,7), (3,5), (3,6)\}$ with $|E(F_{11})| = 12$

Theorem 4. *Equation of elliptic curve over finite field F_p is $E : Y^2 = X^3 + AX + B$. Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be two points on E. If Theorem 2 is applied to points P and Q then the resulting point also lies on E.*

Proof. Theorem 2 is derived by substituting the equation of line to elliptic curve. So the solution automatically satisfies the elliptic curve. Similarly Theorem 4 is also true.

□

Example: Consider the above example. Let $P = (0,1)$ and $Q = (10,0)$. $\lambda = \frac{1-0}{0-10} = \frac{1}{-10} = 1 \pmod{11}$,$x_3 = 1 - 0 - 10 = -9 = 2 \pmod{11}$
$y_3 = 1(0 - (2)) - 1 = -3 = 8$ and $(2,8) \in E$

Theorem 5. *The elliptic curve over finite field along with addition property forms finite group.*

2.4 Elliptic curve discrete logarithm problem

Definition 7. *Let $(G, .)$ be a group and g be the generator of the group and $h \in G$ such that*

$$g^x = h$$

x is called discrete logarithm of h base x and is denoted by $x = log_g h$.

Elliptic curve group operation is addition as described above.So, in case of elliptic curve discrete logarithm problem is described

as follows: Let $P, Q \in E$ such that $nP = Q$. n is called elliptic discrete logarithm of Q base P and is denoted by $n = \log_P Q$.

It should be noted that if Q is not multiple of P, then n will not be defined. However, for all practical purposes, Q is obtained from repeatedly adding P. So n will exist.

We also note that in every finite group every element has finite order. Let order of P be s i.e. $sP = \mathscr{O}$. If n_0 is the smallest number satisfying $Q = sP$. Then for all $n = n_0 + is$ $\forall i \in \mathbb{Z}$, $Q = nP$.It implies $\log_P Q$ is an element of $\mathbb{Z}/s\mathbb{Z}$.We could also set the value to be n_0.But if we define the value to be in $\mathbb{Z}/s\mathbb{Z}$ the the following equation is satisfied.

$$\log_P(Q + R) = \log_P(Q) + \log_P(R)$$

Therefore the function $log_P : E(F_p) \to \mathbb{Z}/s\mathbb{Z}$ defines group homomorphism.

2.4.1 Double and Add Algorithm

The value of n in $Q = nP$ is large for practical application.If nP is calculated by adding P repeatedly n number of times, it will $O(n)$ operation. Suppose n has k binary digits. Then the complexity will be $O(2^k)$. By the use of double and add algorithm we can reduce the complexity to $O(\log n)$ or $O(k)$.
Take $n = 151$. Binary representation of 151 is 10010111_2.

$$151 = 1.2^7 + 0.2^6 + 0.2^5 + 1.2^5 + 0.2^4 + 1.2^3 + 1.2^2 + 1.2^1 + 1.2^0$$

or

$$151P = 2^7P + 2^4P + 2^2P + 2^1P + 2^0P$$

Obtain $151P$ using double and add algorithm

1. Start with P
2. Double it to get $2P$ $(P + P)$

3. Add $2P$ to P (result will be $2^1P + 2^0P$)
4. Double $2P$ to get $4P$ $(2P + 2P)$
5. Add $4P$ to the result (result will be $2^2P + 2^1P + 2^0P$)
6. Double $4P$ to get $8P$ $(4P + 4P)$
7. Double $8P$ to get $16P$ $(8P + 8P)$
8. Add $16P$ to the result (result will be $2^4P+2^2P+2^1P+2^0P$)
9. Keep Doubling $16P$ till $128P$ is obtained
10. Add $128P$ to the result (result will be $2^7P + 2^4P + 2^2P + 2^1P + 2^0P$)

We obtained $151P$ using only four addition and seven doubling operation!

Pseudo-code of double and add algorithm
Let $n = n_0 + 2n_1 + 2^2n_2 + ... + 2^mn_m$ where $n_0, ...n_m \in \{0, 1\}$
$m = \lfloor \log_2 n \rfloor$

1. $bits = [n_0, n_1, ...n_m]$
2. $result = \mathscr{O}$ // infinity point
3. $temp = P$
4. for bit in bits:
5. if bit == 1:
6. result = result + temp // addition operation
7. temp = temp + temp // doubling operation
8. return result

2.4.2 Hasse's Theorem

Hasse's theorem estimates the number of points in an elliptic curve over a finite field. We state Hasse's theorem without proof.

Theorem 6. *Let N is the number of points on an Elliptic Curve E over finite field F_q. Then the following inequality is true*

$$|N - (q+1)| < 2\sqrt{q}$$

or

$$q + 1 - 2\sqrt{q} < N < q + 1 + 2\sqrt{q}$$

2.4.3 How to solve ECDLP

Suppose P and Q are points on an Elliptic Curve E over finite field F_p such the $Q = nP$. The value of P and Q are known and we wish to find the value of n.

Exhaustive Search Method
Compute the value of $P, 2P, 3P...$ until you find a multiple such that it is equal to Q. Time complexity of this algorithm is $O(p)$.

Collision Search Method
Compute two lists for randomly chosen integers $m_1, m_2, ...m_r$ and $n_1, n_2, ...n_r$ where each integer is between 1 and p.
List 1: $m_1P, m_2P, ..., m_rP$
List 2: $n_1P + Q, n_2P + Q, ..., n_rP + Q$
As soon as we find m_uP and m_vP+Q such that $m_uP = m_vP+Q$, then $Q = (m_v - m_u)P$ i.e. $n = m_v - m_u$. Time complexity of collision search method is $O(\sqrt{p})$ by the birthday paradox[2].

The value of p is taken very large for practical purpose. For example, Secp256k1 is the name of the elliptic curve used by Bitcoin to implement its public key cryptography. The value of p used by it is $p = 2^{256}–2^{32}–2^9–2^8–2^7–2^6–2^4–1$.

2.5 Elliptic Diffie–Hellman key exchange

The biggest problem of symmetric cryptography is sharing of private keys. Historically secret key used to be shared physically. After that, public channels were used to share the encrypted message. Diffie-Hellman key exchange helps the sender and receiver of the message to exchange cryptographic keys over a public channel securely. Its security does depend on mathematical principles.

Elliptic Diffie-Hellman Key Exchange Algorithm

1. A large prime number p, an Elliptic Curve E over F_p and a point $P \in E(F_p)$ is shared between Ram and Shayam over public channel.
2. Ram chooses a large random number n_R and computes $Q_R = n_R P$
3. Shayam chooses a large random number n_S and computes $Q_S = n_S P$
4. Ram and Shayam shares Q_R and Q_S over the public channel.
5. Ram now computes $n_R Q_S$ and Shayam computes $n_S Q_R$
6. The shared secret key is $Q = n_R Q_S = n_R(n_S P) = n_S(n_R P) = n_S Q_R$

Suppose Ghanshyam wants to obtain the secret key. Ghanshyam knows the value of P, Q_R, Q_S as they are available on the public channel. Ghanshyam can solve Elliptic Curve Discrete Logarithm Problem, i.e., the equation $Q_R = n_R P$ or $Q_S = n_S P$ to obtain the value of n_R and n_S respectively. After the value of Q can be obtained using the formula $Q = n_R Q_S$ or $Q = n_S Q_R$.

However, we know that the best-known algorithm to solve EDCLP has time complexity $O(\sqrt{p})$. If p is chosen very large, then it is not practically feasible to solve EDCLP.

Example Ram and Shayam want to exchange a secret key using the Diffie-Hellman Algorithm with the following curve, prime and point. $p = 7$ $E : y^2 = X^3 + 1$ $P = (2, 3) \in E(F_7)$
Ram and Shayam choose their random secret key as $n_R = 5$ and $n_S = 3$.
Ram computes $Q_R = n_R P = 5(2, 3) = (2, 4) \in E(F_7)$
Shayam computes $Q_S = n_S P = 3(2, 3) = (6, 0) \in E(F_7)$
Ram and Shayam will share the value of Q_R and Q_S with each other.
Ram computes $n_R Q_S = 5(6, 0) = (6, 0)$
Shayam computes $n_S Q_R = 3(2, 4) = (6, 0)$
Ram and Shayam have shared the secret key $(6, 0)$. They will use x-coordinate as the shared key and ignore y-coordinate.

2.6 ElGamal Elliptic Curve Cryptosystem

ElGamal Elliptic Curve Cryptosystem is an example of asymmeteric key cryptography. It is similar to Diffie-Hellman key exchange algorithm but is used to securely exchange messages over public channel. Now let's have a look on the example where Shayam wants to share a message with Ram using ElGamal Elliptic Curve Cryptosystem.

ElGamal Elliptic Curve Cryptosystem

1. A large prime number p, an Elliptic Curve E over F_p and a point $P \in E(F_p)$ is shared between Ram and Shayam over public channel.

2. Ram chooses a large secret random number n_R and computes $Q_R = n_R P$. Q_R is Ram's public key.

3. Shayam chooses a plain text M such that $M \in E(F_p)$.

4. Shayam chooses a transitory key $k \in \mathbb{Z}$ and computes and shares X_1 and X_2 with Ram

$$X_1 = kP$$

$$X_2 = M + kQ_R$$

5. Ram will obtain M from n_R, X_1 and X_2 using the formula given below

$$X_2 - n_R X_1 = M + kQ_R - n_R(kP) = M$$

One of the drawback of this method is to attach plain text to a point in $E(F_p)$. One of the proposed solution for this is to use probablistic algorithm [1] for this with a small chance of failure.

Chapter 3

Implementation

This chapter deals with the implementation of ECC using javascript.

```
INF_POINT = null;
class EllipticCurve {
  constructor(a, b, p) {
    this.a = a;
    this.b = b;
    this.p = p;
    this.points = [];
    this.definePoints();
  }

  definePoints() {
    this.points.push(INF_POINT);
    for (let x = 1; x < this.p; x++) {
      for (let y = 1; y < this.p; y++) {
        if (this.equalModP(y ** 2, x ** 3 + this.a * x + this
   .b)) {
          this.points.push([x, y]);
        }
      }
    }
  }

  add(p, q) {
    if (p == null) {
      return q;
    }
    if (q == null) {
      return p;
    }
```

```
29
30      const x1 = p[0];
31          const y1 = p[1];
32          const x2 = q[0];
33          const y2 = q[1];
34      let lambda;
35      if (this.equalModP(x1, x2) && this.equalModP(y1, -y2)) {
36        return INF_POINT;
37      }
38      if (this.equalModP(x1, x2) && this.equalModP(y1, y2)) {
39        lambda = this.reduceModP((3 * x1 * x1 + this.a) * this.
    inverseModP(2 * y1));
40      } else {
41        lambda = this.reduceModP((y1 - y2) * this.inverseModP(
    x1 - x2));
42      }
43      const x3 = this.reduceModP(lambda ** 2 - x1 - x2);
44      const y3 = this.reduceModP(lambda * (x1 - x3) - y1);
45      return [x3, y3];
46    }
47
48    doubleAndAddAlgorithm(n, P) {
49      let Q = INF_POINT;
50      while (n != 0) {
51        if (n & 1 !== 0) {
52          Q = this.add(Q, P);
53        }
54        P = this.add(P, P);
55        n >>= 1;
56      }
57      return Q;
58    }
59
60    reduceModP(n) {
61      while (n < 0) {
62        n += this.p;
63      }
64      return n % this.p;
65    }
66
67    equalModP(x, y) {
68      return this.reduceModP(x - y) == 0;
69    }
70
71    inverseModP(x) {
72      for (let y = 1; y < this.p; y++) {
73        if (this.equalModP(x * y, 1)) {
74          return y;
75        }
```

```
76      }
77      return null;
78    }
79  }
80  ec = new EllipticCurve(0, 1, 7);
81  console.log(ec.points);
82  console.log(ec.doubleAndAddAlgorithm(5, [2, 4]));
```

Bibliography

[1] Debabrat Boruah and Monjul Saikia. Implementation of elgamal elliptic curve cryptography over prime field using c. In *International Conference on Information Communication and Embedded Systems (ICICES2014)*, pages 1–7, 2014.

[2] Jeffrey Hoffstein, Jill Pipher, Joseph H Silverman, and Joseph H Silverman. *An introduction to mathematical cryptography*, volume 1. Springer, 2008.

[3] Wikipedia contributors. Cryptography — Wikipedia, the free encyclopedia, 2022. [Online; accessed 29-April-2022].