

ABSTRACT

This study describes three algorithms for efficient implementations in Elliptic Curve Cryptography (ECC). The first algorithm determines an approach of performing key exchanges between two subgroups for Decomposition Problem and three subgroups for Triple Decomposition Problem. The algorithms work by arranging parameters using finite field group in elliptic curve E . It is a new approach which performs core operation using multiplication of points based in ECC. The algorithm explores computational advantages of computing cofactor number of points on E and it is computationally infeasible to obtain if the cofactor are large enough. This approach presents better platform in finite field E as compared to the original works using the braid groups. The second algorithm deals with the use of Decomposition Problem in encryption scheme for ECC. We introduce two concepts of splitting messages using the scheme in El-Gamal and Massey-Omura algorithms. The messages can be split either before or after the user sends the messages to the receiver. The third algorithm describes the application of Decomposition Problem to the signing and verifying digital messages in ECC. Since subexponential-time algorithm is known for ordinary discrete logarithm problem and integer factorization problem and not for elliptic curve discrete logarithm problem, the algorithm presented for the digital signature in this study has substantially greater strength per key bit than in other digital signature algorithm.

ABSTRAK

Kajian ini menerangkan tentang tiga jenis algorithma yang digunakan dalam penggunaan kaedah Kriptografi Lengkung Elliptik (ECC). Algorithma pertama mengenai protokol pertukaran kekunci antara dua sub kumpulan menggunakan kaedah Masalah Penguraian dan tiga sub kumpulan menggunakan kaedah Masalah Penguraian Gandaan Tiga. Algorithma baru ini berfungsi dengan mengubah parameter menggunakan medan finit dalam lengkung elliptik E . Ia menggunakan operasi asas pendaraban titik berdasarkan ECC. Ia juga menunjukkan manfaat pengiraan kofaktor nombor dalam E yang mana ia adalah tak tersaur apabila melibatkan kofaktor yang lebih besar. Pendekatan ini juga memberi platform yang lebih baik dari kaedah asal yang menggunakan kumpulan Braid. Algorithma kedua pula menghasilkan skema enkripsi berdasarkan kaedah Masalah Penguraian bagi ECC. Dua konsep diperkenalkan untuk memisahkan mesej dengan menggunakan skema ElGamal dan Massey Omura. Mesej tersebut boleh dipisahkan sebelum atau selepas pengguna menghantar mesej kepada penerima. Algorithma ketiga pula mengkaji penggunaan kaedah Masalah Penguraian untuk tandatangan digital dalam ECC. Memandangkan algorithma subeksponen-masa dikenali untuk masalah diskrit log biasa dan masalah pemfaktoran integer dan bukannya untuk masalah diskrit logaritma bagi lengkung elliptik, maka algorithma yang diperkenalkan untuk tandatangan digital dalam kajian ini lebih kukuh penggunaannya berbanding yang terdahulu.

ACKNOWLEDGEMENTS

First, I thank to Allah for His blessing in granting me to complete this studies.

I want to express my gratitude to Assoc. Prof. Dr. Wan Ainun Mior Othman for her supervision, enthusiasm and important contribution to this study. Her precious ideas and advices were the great practical throughout the writing of this thesis.

My deepest gratitude and appreciation to my father, Mr. Zazali Chik and my mother, Mrs. Zaliha Abd Wahab for their endless love, prayers and full support for me to get through all the obstacle on completing this study. Thank you for always been there for me. Also thank to my beloved sisters and brother, Azyyati Adiah, Muhammad Iznán and Amirah Afiqah, even though I am facing hard time in this journey, do not ever make it discourage all of you to pursue your own dreams.

I dedicate my special appreciation to my dearest husband, Mr. Fatos Omar Othman, for always encourage me to be myself regardless of what confronts me and always pushing me up when my motivation had away. It must be more difficult for me to complete this study without you.

And thank to Idayu, KuAzlina, Norli, Iqbal, Akma, Rashidah, Suazlan, Adzhar, Jia Hou, Siew Kien and all of my ISM friends for your supports and contribution of ideas in this studies. Also thanks for helping me up in L^AT_EX and Matlab.

Hilyati Hanina Zazali

February 2012

PUBLICATIONS

1. Zazali, H.H., & Othman, W.A.M. (2012). Key Exchange in Elliptic Curve Cryptography based on Decomposition Problem. *Sains Malaysiana* 41(7), pp. 907-910. UKM: Malaysia. ISI INDEXED
2. Zazali, H.H., & Othman, W.A.M. (2009a). New Key Exchange in Elliptic Curve based on the Decomposition Problem. *Proceeding of The 5th International Conference on Mathematics, Statistic and their applications*. Padang: Indonesia.
3. Othman, W.A.M., & Zazali, H.H. (2009b). Key Exchange Method using Triple Decomposition Problem in Elliptic Curve. *Proceeding of Simposium Kebangsaan Sains Matematik (SKSM)* (pp. 913 - 916). Melaka: Malaysia.

PRESENTATIONS AND SEMINARS

1. 5th International Conference on Mathematics, Statistics and Their Applications (ICMSA). (June 2009). Padang: Indonesia.
2. Simposium Kebangsaan Sains Matematik (SKSM). (December 2009). Melaka: Malaysia.
3. Applied Mathematics International Conference (AMIC). (June 2010). Kuala Lumpur: Malaysia.

CONTENTS

	Page
ABSTRACT	ii
ABSTRAK	iii
ACKNOWLEDGEMENTS	iv
PUBLICATIONS	v
PRESENTATIONS AND SEMINARS	vi
CONTENTS	ix
LIST OF TABLES	x
LIST OF FIGURES	xi
ABBREVIATIONS	xii
1 INTRODUCTION	1
1.1 Literature Review	1
1.2 Objective of the Research	4
1.3 Scope of the Research	5
1.4 Thesis Organisation	6
2 MATHEMATICAL BACKGROUND	8
2.1 Elliptic Curve Cryptography	9
2.1.1 Mathematical background in Elliptic curves	10
2.1.2 Algebraic structure in Elliptic curve	11
2.1.3 The computations in Elliptic curve	15

2.1.4	Choosing points on the curve, $\#E(F_q)$	17
2.1.5	Discrete Logarithm Problem on Elliptic Curves	18
2.1.6	Example on computing addition and doubling point on E	18
2.2	Summary	21
3	KEY AGREEMENT	22
3.1	Introduction to Key Agreement	22
3.2	Decomposition Problem in Elliptic Curve Cryptography	26
3.2.1	Summary of DPECC	29
3.2.2	Example of DPECC	30
3.3	Triple Decomposition Problem in Elliptic Curve Cryptography	33
3.3.1	Summary of TDPECC	35
3.3.2	Example of TDPECC	36
3.4	Implementation issues	40
4	ENCRYPTION IN CRYPTOGRAPHY	42
4.1	Introduction to Encryption	42
4.2	El-Gamal encryption based on DPECC	46
4.2.1	Example of El-Gamal encryption based on DPECC	48
4.3	Massey-Omura encryption based on DPECC	49
4.4	Implementation issues	50
5	DIGITAL SIGNATURE SCHEME	51
5.1	Introduction to Digital Signature Algorithm	51
5.2	Decomposition Problem in Elliptic Curve Cryptography in Digital Signature Algorithm	53
5.3	Summary of DPECC in Signing Digital Messages	57

5.4	Implementation Issues	59
6	CONCLUSION	60
A	Elliptic Curves	62
A.1	Weierstrass equation	62
A.1.1	Simplified Weierstrass equation	63
A.2	Group Law	65
A.2.1	Addition and Doubling operation	65
A.3	Group order	66
A.4	Number theory	67
A.5	Group Theory	69
A.6	Field theory	73
A.6.1	Finite Fields	76
B	Programming code	77
B.1	The Matlab code for addell.m	78
B.2	The Matlab code for multell.m	82
B.3	The Matlab code for multsell.m	83
B.4	The Matlab code for randprime.m	84
	REFERENCES	85

LIST OF TABLES

	Page
2.1 Group	13
2.2 Ring	14
2.3 Field	15
3.1 Summary of DPECC	29
3.2 Summary of TDPECC	35
5.1 Summary of DPECC in Signing and Verifying Digital Messages	57

LIST OF FIGURES

	Page
2.1 Asymmetric Cryptography	8
2.2 Elliptic Curve E	16
2.3 Elliptic curve $\mathbf{GF}(29)$	19
2.4 All the computation points on $\mathbf{GF}(29)$	21
3.1 Elliptic Curve defined by $y^2 = x^3 + 8x + 1$ over Finite Field of size 101	31
3.2 Elliptic Curve defined by $y^2 = x^3 + 24x + 13$ over Finite Field of size 29	37
5.1 DPECC Signing	58
5.2 DPECC Verifying	58
A.1 Elliptic Curves examples on \mathbf{R}	63
A.2 The mathematical concept of Algebraic Structure	72

ABBREVIATIONS

ANSI	-	American National Standard Institute
DHKE	-	Diffie-Hellman Key Exchange
DP	-	Decomposition Problem
DPECC	-	Decomposition Problem in Elliptic Curve Cryptography
DSS	-	Digital Signature Standard
DSA	-	Digital Signature Algorithm
EC	-	Elliptic Curve
ECC	-	Elliptic Curve Cryptography
ECDLP	-	Elliptic Curve Discrete Logarithm Problem
ECDSA	-	Elliptic Curve Digital Signature Algorithm
FIPS	-	Federal Information Processing Standard
FR	-	field representation
GCD	-	Greatest Common Divisor
GF	-	Galois Field
IBM	-	International Business Machines
ISO	-	International Organization for Standardization
NIST	-	National Institute of Standards and Technology
RSA	-	Riverst-Shamir-Adleman
SHA-1	-	Secure Hash Algorithm
TDP	-	Triple Decomposition Problem
TDPECC	-	Triple Decomposition Problem in Elliptic Curve Cryptography

CHAPTER 1

INTRODUCTION

Cryptography becomes very essential since the demand for the information interchange and electronic services has been increasing with the alarming needs for secured data. In order to provide safe operation for transaction of valuable information stands a strong mathematical theory for cryptography.

The main focus in this study is to create suitable key exchange for use by cryptography. The description and implementation of mathematics in key exchange help to understand how the system works in practice. For securely transmitting messages, key exchange will be presented in a process called encryption, which is a process of changing message into some different characters that cannot be unreadable. Then user needs to sign the documents using digital signatures to make sure trust among the transaction.

The algorithm which is used in encryption is called cryptographic algorithm and the systems that implement such algorithm are called cryptosystems. There are two types of cryptosystems that implement cryptographic algorithms. They are known as Asymmetric cryptosystem which is a combination of public and private key and Symmetric cryptosystem which only used private key.

1.1 Literature Review

The first concept of Asymmetric cryptography was first introduced by Whitfield Diffie and Martin Hellman (Diffie and Hellman, 1976). Since then, the study of public-key cryptography has grown rapidly. In 1977, Rivest, Shamir and Adleman invented the well known RSA public-key cryptosystem (Rivest et al., 1978). Most of

the cryptography systems focus on finite fields. The finite field of the form \mathbf{F}_q , where q is a prime number, can be used to implement public-key cryptography algorithms. The Diffie-Hellman (DH) key agreement was the first and the best known example for public-key cryptography.

There are two types of finite fields that are popular in cryptography operations which is a prime Galois Fields ($GF(p)$) and binary extension Galois Fields ($GF(2^m)$). The basis of Galois Field (GF) related operations are integer modular arithmetic operations which consists of basic operation in modular inversion, modular division, modular multiplication, and modular addition or subtraction operations.

Over past 150 years before, elliptic curves have been studied extensively as algebraic and geometric entities and from these studies has emerged a rich and deep theory. In 1985, Elliptic Curve Cryptography (ECC) in Asymmetric cryptosystem was first discovered independently by Neal Koblitz from University of Washington and Victor S. Miller from IBM. ECC been introduced as a group of points on an elliptic curve over finite field which can be used for encrypting data and provides more security than the fields of the form \mathbf{F}_q . Since then, elliptic curves have played a significant role in public-key cryptography. ECC can provide the same level of security as RSA cryptosystem with much smaller key size. For example, a 160-bit ECC is as secured as 1024-bit RSA cryptosystem (Rivest et al., 1978). The use of smaller keys gives computationally more efficient algorithm for cryptosystems as compared to the traditional cryptographic algorithms.

ECC presents its wide use in various public-key cryptography algorithms, particularly involving discrete logarithms such as:

1. Elliptic Curve Diffie-Hellman (ECDH) which is the key exchange method in EC.

2. Elliptic Curve El-Gamal Cryptosystem which is the famous encryption scheme in EC.
3. El-Gamal Digital Signature which is the signing and verification method in EC.

As in this research, the main focus will be on the study of new cryptographic algorithms for key agreement scheme in ECC. A well-known key exchange in elliptic curve is ECDH, which is based on the additive elliptic curve group. Originally the idea is an implementation of the famous Asymmetric cryptosystem based on the multiplicative group modulo p , Diffie-Hellman algorithm (Diffie and Hellman, 1976). In 2005, we have been introduced by the idea of key exchange protocol based on Decomposition Problem (DP) (Shpilrain and Ushakov, 2005). It is a presentation of the decomposition problem in non-commutative group which involves two subgroups over the main platform. The subgroups may contain different elements for each user. But, the presentation of the platform in this method seems to be more complicated for implementation purposes. The elements need to denote by certain centralizers on a platform group G which has to be non-commutative. A year later, another extension method has been developed known as Triple Decomposition Problem (TDP) (Kurt, 2006). TDP is a presentation of exchanging key into three different subgroups as the main platform. By adopting these two ideas, we use ECC as the main platform and introduced it as Decomposition Problem in Elliptic Curve Cryptography (DPECC) (Zazali and Othman, 2009) and Triple Decomposition Problem in Elliptic Curve Cryptography (TDPECC) (Othman and Zazali, 2009).

We continue the study of developing algorithm based on the idea of DPECC into encryption method. First we study the encryption in El-Gamal (1985) and

Massey-Omura (1983) for ECC, and do some comparison on how the algorithm may works by splitting the message into n shares. Based on the ideas, we developed Encryption using Decomposition Problem in Elliptic Curve Cryptography. The working algorithm in TDPECC for encryption will not be discussed in this thesis due to the time constraint. Further researches are needed to make sure the process of implementation of TDPECC in encryption may works perfectly before proceeds to signing the digital messages.

To make it useful for security, we complete the study with the implementation of DPECC in the digital signature. A signed of electronic documents is important to represent trust relationships. We study the previous protocol for digital signature in ECC known as Elliptic Curve Digital Signature Algorithm (ECDSA) (Vanstone, 1992) and developed our own digital signature called Digital Signature based on DPECC.

Useful features of ECC gives the utmost benefits to the method. The information used to implement the code can be shared in public by the users who wish to communicate, without deciphering key. This, has been known to others, which makes it unnecessary to have a private meeting to agree upon such keys, and makes the codes workable in the context of electronic communication. The method has been widely used especially over the internet and wireless systems, where eavesdropping is often possible.

1.2 Objective of the Research

This thesis explores the study of elliptic curves over the finite field method. Capabilities of providing shorter key lengths compare to traditional methods help in squeezing the cryptosystem to the limited environment for real world application.

In 2005, Decomposition Problem based on discrete logarithm protocol has been introduced by Vladimir Shpilrain and Alexander Ushakov. It is a method of developing key exchange that involves non-commutative group to create two subgroups containing different elements for each user. And in 2006, Yaşem Kurt produced an extension method from DP key exchange that creates three unknown subgroups named as Triple Decomposition Problem.

Intensive studies of both methods have shown that it may have potential use in elliptic curves. Therefore, we proposed a new platform for both methods in elliptic curve finite field and performs the methods in three basic studies in cryptography which consist of key exchange, encryption and digital signature.

The objectives for each application will be represented as follows:

1. To design an alternative method of key exchange using Decomposition Problem and Triple Decomposition Problem based on elliptic curves in cryptography.
2. To design an encryption method in ECC based on proposed method in key exchange.
3. To design a trusted certification in digital signature for ECC from proposed protocol in key exchange and encryption.

1.3 Scope of the Research

The point of view offered in the research consists of the study of applied cryptography in mathematics, engineering, physics and computer science. The study begins with the basic arithmetic operation in elliptic curves over finite field. Together with that, we study the algorithms, architectures and the implementation of elliptic curve in cryptography, which relates to the main parts known as key exchange, encryption and digital signature. Therefore, the implementations must consist of the

security services, authentication, confidentiality, data integrity and non-repudiation specifically for ECC finite field. We continue the study in significant Asymmetric cryptosystem over discrete logarithm based method in Decomposition Problem (Shpilrain and Ushakov, 2005) and Triple Decomposition Problem (Kurt, 2006). From both cryptosystem, we developed new idea based on ECC as the main platform. At the end of the study, we test the effectiveness of the system in Matlab and Sage software, and present it in examples to ensure the reliability of the study.

1.4 Thesis Organisation

This thesis is organized as follows. It consists of 6 chapters.

Chapter 2 introduces the basic mathematical background which needed to understand the concept followed in this thesis. This chapter highlights on the concept of arithmetic operations in ECC, the discrete logarithm problem on an elliptic curve and some of its properties which is compatible to be used in Decomposition Problem and Triple Decomposition Problem.

Chapter 3 introduces the key agreement scheme and the algorithm involve in the scheme. This chapter is a major part as it introduces the main research in our study. We started with the study of the first and well-known Asymmetric cryptosystem which is Diffie-Hellman Key Exchange (DHKE) scheme (1976). We continue the study of key exchange in cryptography from Decomposition Problem (DP) (Shpilrain & Ushakov, 2005) and Triple Decomposition Problem (TDP) (Kurt, 2006). Motivated from these schemes, we designed the protocols of DP and TDP by using ECC as the platforms. To strengthen the study, we included examples and calculations using Matlab. For the next chapters, we continue our key exchange study from DP in ECC to be apply in Encryption and Digital Signature schemes.

TDP schemes in ECC will not be discussed through the next chapters, because it will be conducted for the future research.

Chapter 4 introduces the study of encryption involves in ECC. Encryption using El-Gamal will be the basic study for creating the new alternative method. It is based on DPECC that has been mentioned in the previous chapter. A full algorithm for encryption and decryption will be included.

Chapter 5 illustrates the study of signing electronic messages for ECC. From the basic idea of signing messages called Elliptic Curve Digital Signature Algorithm (ECDSA), we show how we manage to use the main idea of key exchange based on DPECC, into signing and verifying the digital messages.

Chapter 6 explains the overall works and contributions of the study in cryptography and elliptic curves and also the recommendation for future work.

CHAPTER 2

MATHEMATICAL BACKGROUND

Asymmetric cryptosystem also known as public-key cryptography which involves the use of secret-key with an addition of public-key. It is an improvement from the traditional Symmetric cryptosystem which allows two parties to exchange data privately in the presence of possible eavesdroppers, without previously agreeing on a shared secret. Figure 2.1 shows how asymmetric-key is based on a matched cryptographic key pair, which is the key split into two different keys, the private-key and public-key.

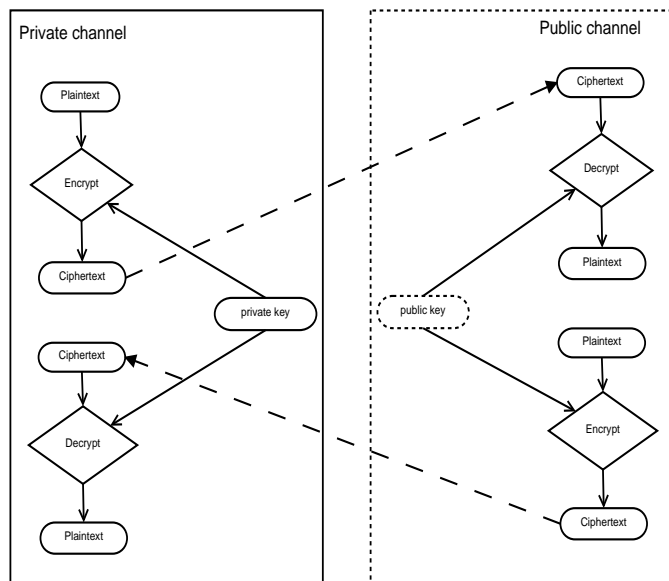


Figure 2.1: Asymmetric-key Cryptography

The implementation of algorithms in Asymmetric cryptosystem make the works much slower than Symmetric cryptosystem, but it is widely used nowadays because of the security relies on it. One of the famous studies in Asymmetric cryptosystem is on Elliptic Curve Cryptography(ECC). Elliptic curves are not new in the study of number theories, but the application of it is still recent. In 1985, Professor Neal

Koblitz, a mathematician from University of Washington, and Dr. Victor Miller, a scientist from IBM, discovered a new system of cryptography based on elliptic curves. The mathematical implementation using ECC is difficult to be broken, but the application of this study is easy to implement. Therefore, the main focus in this study will be on the study of ECC and their applications to cryptosystems. The difficulties of the computation in ECC depends on the abilities of taking two points on the specific curve over finite fields, applies the addition method between them, and gets another point on the same curve. This chapter summarizes the previous works in mathematical concept of ECC to help understand the whole cryptosystem study.

2.1 Elliptic Curve Cryptography

ECC is a cryptosystem method which utilizes points on elliptic curves. These points can be represented graphically in a two dimensional-plane, or a toroid (Stalling, 2006; Schneier, 1996). Elliptic curves are formed from a cubic equation in two variables called Weierstrass equation:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (2.1)$$

The mathematics associated in variable x and y can be real, complex, integers, polynomial basis, optimal normal basis or any kinds of field element (Rosing, 1998). In this thesis, we are dealing more to the real numbers as examples and understanding of the fundamentals of algorithms. Galois field (**GF**) over prime field (p) and binary field (2^m) are specific condition to perform with ECC.

For finite prime field, **GF**(p) of order p , utilizes equation 2.1 as:

$$y^2 \equiv x^3 + ax + b \pmod{p} \quad \text{where } 4a^3 + 27b^2 \neq 0 \quad (2.2)$$

and, for finite field $\mathbf{GF}(2^m)$, the equation 2.1 denoted as:

$$y^2 + xy \equiv x^3 + ax^2 + b \pmod{2^m} \quad \text{where } b \neq 0 \quad (2.3)$$

Finite field is important because the study in this research employs operations performed in finite field. Choosing the suitable finite field for ECC gives advantages for the users to generate number of points on the curve (Rosing, 1998).

2.1.1 Mathematical background in Elliptic curves

In this section, we define the basic terminology of creating the curve, and how the computations do involve in ECC. An elliptic curve over finite field of \mathbf{F} is the set of all solutions (known as points) of (x, y) where $x \in \mathbf{F}$ and $y \in \mathbf{F}$, to an equation of a special form, $y^2 = x^3 + ax + b$. In this section, we will proceed to the rules of constructing the curves follow by certain properties and the mathematical background for ECC.

To construct an elliptic curve E , we need to define the curve over a finite field of \mathbf{F} . Since there are finitely pairs of (x, y) over the finite field \mathbf{F} , it shows that the group of $E(\mathbf{F})$ is finite. Finite fields are therefore denoted as $\mathbf{GF}(q)$ where $q = p^n$. The properties of elliptic curve E over finite field are as follows (Washington, 2003):

Theorem 2.1. *Let E be an elliptic curve over the finite field \mathbf{F}_q . Then $E(\mathbf{F}_q) \simeq \mathbf{Z}_n$ for some integer $n \geq 1$ or $\mathbf{Z}_{n_1} \oplus \mathbf{Z}_{n_2}$ for some integer $n_1, n_2 \geq 1$ with n_1 dividing n_2 .*

Theorem 2.2. (Hasse) *Let E be an elliptic curve over the finite field \mathbf{F}_q . Then the order of $E(\mathbf{F}_q)$ satisfies $|q + 1 - \#E(\mathbf{F}_q)| \leq 2\sqrt{q}$.*

Proof: See Appendix A.

Theorem 2.3. *Let $q = p^n$ a power of a prime p and let $N = q + 1 - a$. There is an elliptic curve E defined over \mathbf{F}_q such that $\#E(\mathbf{F}_q) = N$ if and only if $|a| \leq 2\sqrt{q}$ and a satisfied one of the following:*

1. $\gcd(a, p) = 1$
2. n is even and $a = \pm 2\sqrt{q}$
3. n is even, $p \not\equiv 1 \pmod{3}$, and $a = \pm 2\sqrt{q}$
4. n is odd, $p = 2$ or 3 , and $a = \pm p^{(n+1)/2}$
5. n is even, $p \not\equiv 1 \pmod{4}$ and $a = 0$
6. n is odd and $a = 0$.

Theorem 2.4. *Let N be an integer that occurs as the order of an elliptic curve over a finite field \mathbf{F}_q as in Theorem 2.3.*

When $N = p^e n_1 n_2$ with $p \nmid n_1 n_2$ and $n_1 \mid n_2$ (possibly $n_1 = 1$). There is an elliptic curve E over \mathbf{F}_q such that $E(\mathbf{F}_q) \simeq \mathbf{Z}_{p^e} \oplus \mathbf{Z}_{n_1} \oplus \mathbf{Z}_{n_2}$ if and only if:

1. $n_1 \mid q - 1$ in cases (1),(3),(4),(5),(6) of Theorem 2.3
2. $n_1 = n_2$ in case (2) of Theorem 2.3

These are the only groups that occur as groups $E(\mathbf{F}_q)$.

2.1.2 Algebraic structure in Elliptic curve

To determine numbers of points on elliptic curve E over finite field, it must satisfies the combination of the set and operations called algebraic structure. The following are the fundamental definitions that satisfy the algebraic structure in elliptic curve: *groups* and *rings*. Additional information in algebraic structure can be found in (Forouzan, 2008).

Definition 2.1. (Group) A group (\mathbf{G}) is sets of elements with binary operation \bullet , that satisfies four properties (axioms). A **commutative group**, also known as **abelian group**, is group in which operator satisfies the four properties for groups plus property in commutativity. The properties are defined as follows:

Consider an elliptic curve E with points $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$ and $R = (x_R, y_R)$.

Closure: If P and Q are elements of \mathbf{G} , then $R = P \bullet Q$ is also an element of \mathbf{G} . This means that the operation on any two elements in set is another element in the set.

Associativity: If P , Q and R are elements of \mathbf{G} , then $(P \bullet Q) \bullet R = P \bullet (Q \bullet R)$. It does not matter which order it apply the operation.

Commutativity: For all P and Q in \mathbf{G} , then $P \bullet Q = Q \bullet P$.

Note: this only satisfies for a commutative group.

Identity: For all P in \mathbf{G} , there exist an element e , called the identity element, such that $e \bullet P = P \bullet e = P$.

Inverse: For each P in \mathbf{G} , there exist an element P^* , called the inverse of P , such that $P \bullet P^* = P^* \bullet P = e$.

Theorem 2.5. The addition points of an elliptic curve E satisfies the following properties:

1. Commutative $P + Q = Q + P$ for all P, Q on E .
2. Existence of identity $P + \infty = P$ for all points P on E .
3. Existence of inverse Given P on E , there exist P^* on E with $P + P^* = \infty$.

This point P^* will usually denoted $-P$.

Table 2.1: Abelian Group

Algebraic structure	Operation (+ -)
Abelian Group $\mathbf{G} = \langle (+ -) \text{ or } (\times \div) \rangle$	<ol style="list-style-type: none"> 1. Closure 2. Associativity 3. Commutativity 4. Identity 5. Inverse

4. Associativity $(P + Q) + R = P + (Q + R)$ for all P, Q, R on E .

In other words, the points on E form an additive abelian group with ∞ as the identity element.

Definition 2.2. (*Cyclic group, Group Generator*) A group \mathbf{G} is said to be cyclic if there exists an element $a \in \mathbf{G}$ such that for any $b \in \mathbf{G}$, there exists an integer $i \geq 0$ such that $b = a^i$. Element a is called generator of \mathbf{G} . \mathbf{G} is also called the group generated by a . When a group is generated by a , we can write $\mathbf{G} = \langle a \rangle$. Details can be referred at Mao (2003).

Definition 2.3. (Ring) A ring, denoted as \mathbf{R} with two binary operations; addition/subtraction (+ / -) and multiplication (\bullet). For the first operation must satisfy all five properties required for abelian group. The second operation must be satisfied by the following:

Consider an elliptic curve E with points $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$ and $R = (x_R, y_R)$.

Closure: If $P, Q \in \mathbf{R}$ then $R = P + Q$ where $R \in \mathbf{R}$

Associativity: \mathbf{R} is an abelian group with respect to addition. If $P, Q, R \in \mathbf{R}$ with $P + (Q + R) = (P + Q) + R$

Commutativity: For all $P, Q \in \mathbf{R}$ where $P \bullet Q = Q \bullet P$

Distributivity: For all $P, Q, R \in \mathbf{R}$ where $P(Q + R) = PQ + PR$

Table 2.2: Ring

Algebraic structure	Operation (+ -)	Operation ($\times \div$)
Ring	1. Closure	1. Closure
$\mathbf{R} = \langle (+ -) \text{ and } (\times) \rangle$	2. Associativity	2. Associativity
	3. Commutativity	3. Commutativity
	4. Identity	4. Distributivity
	5. Inverse	

A field is a commutative ring that satisfies the elements \mathbf{F} form an abelian group under the operation addition (+) with 0 as the identity element. The rest of the elements of \mathbf{F} other than the ones that form the additive associativity form an abelian group under the operation multiplication (\bullet) with 1 as the identity element. The distributive law holds for the two binary operations such that for all $a, b, c \in \mathbf{F}$, $a(b + c) = (a \bullet b) + (a \bullet c)$. If the number of elements is finite, the field is called a Finite Field. The field with p^n elements is called $\mathbf{GF}(p^n)$. For every power p^n of a prime, there exists exactly one finite field with p^n elements, and these are the only finite fields.

Definition 2.4. (Field) A field is denoted by \mathbf{F} , is a commutative ring with second operation satisfies all five properties defined for first operation except the identity of the first operation has no inverse.

Table 2.3: Field

Algebraic structure	Operation (+ -)	Operation ($\times \div$)
Field $\mathbf{F} = \langle (+ -) \text{ and } (\times \div) \rangle$	1. Closure 2. Associativity 3. Commutativity 4. Identity 5. Inverse	1. Closure 2. Associativity 3. Commutativity 4. Identity 5. Inverse

Definition 2.5. (*Finite Field*) A finite field, is a field that contains a finite number of elements, also known as Galois Fields¹ and denoted by $\mathbf{GF}(q)$.

2.1.3 The computations in Elliptic curve

In this section, we discuss the theories of elliptic curves over \mathbf{F}_q or $\mathbf{GF}(q)$, where q is prime with characteristic greater than 3 ($q > 3$). We defined elliptic curve by an equation of the form $y^2 = x^3 + ax + b$ where $a, b \in \mathbf{F}_q$ and $4a^3 + 27b \neq 0 \pmod{q}$. Notice that for all points (x, y) , there will be $x \in \mathbf{F}_q$, $y \in \mathbf{F}_q$ and point at infinity. From Figure 2.2, let say $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$ are points on the curve, E given by the equation $y^2 = x^3 + ax + b$. The addition from P and Q produce the third point R . A line, L will be drawn through the points P and Q . The intersect line on E , will be the third point as R^* . The reflection of this point across to the x -axis will obtain the R point. Which is $P + Q = R = (x_R, y_R)$.

For elliptic curves computation, users need to know that there will be different operations on computing the addition and double points.

¹introduced by Evariste Galois, a French mathematician in 1830 in his proof of insolvability of the general quintic equation

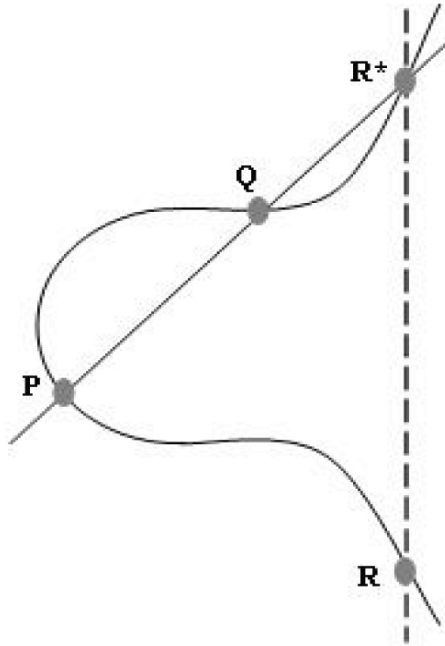


Figure 2.2: Elliptic Curve E

- (a) Assume that $P \neq Q$, and neither points is ∞ . Draw a line, L through P and Q . The slope is

$$m = \frac{y_Q - y_P}{x_Q - x_P}$$

If $x_P = x_Q$, then L will be vertical. But here, just assume that $x_P \neq x_Q$. Then, equation of L will be $y = m(x - x_P) + y_P$. Substitute $(m(x - x_P) + y_P)^2 = x^3 + ax + b$. And arrange it as the form $0 = x^3 - m^2x^2 + \dots$.

The three roots of this cubic correspond to the three points of intersection of L with E . Then we obtain $x = m^2 - x_P - x_Q$, $y = m(x - x_P) + y_P$.

The reflection across the x -axis obtain $R = (x_R, y_R)$, given by

$$\begin{aligned} x_R &= m^2 - x_P - x_Q \\ y_R &= m(x_P - x_R) - y_P \end{aligned}$$

- (b) For $x_P = x_Q$ and $y_P \neq y_Q$. The line through P and Q is a vertical line, and

intersect, E at ∞ . Reflecting ∞ across the x -axis yields the same point ∞ . Therefore, this case $P + Q = \infty$.

Consider the case where $P = Q = (x_P, y_P)$. When two points on a curve are very close to each other, the line through them approximates a tangent line. Let, line L as the tangent line through this two points. We determine the slope m of L using implicit differentiation as below:

$$\begin{aligned} 2y \frac{dy}{dx} &= 3x^2 + A, \text{ so} \\ \frac{dy}{dx} &= \frac{3x_P^2 + A}{2y_P} \\ &= m \end{aligned}$$

(c) If $y_P = 0$, then the line is vertical and $P + Q = \infty$ as before. Therefore, assume that $y_P \neq 0$. The equation of L is $y = m(x - x_P) + y_P$ as before. We obtain the cubic equation $0 = x^3 - m^2x^2 + \dots$. In this case, it will be double root since L is tangent to E at P . Therefore, we obtain

$$\begin{aligned} x_R &= m^2 - 2x_P, \\ y_R &= m(x_P - x_R) - y_P \end{aligned}$$

(d) Suppose that $P_Q = \infty$. The line through P and ∞ is a vertical line that intersect E in the point P^* , and the reflection of P across the x -axis. When reflect the P^* across the x -axis get $R = P + Q$, so we are back at the P . Therefore, $P + \infty = P$ for all points P on E .

2.1.4 Choosing points on the curve, $\#E(F_q)$

In this section, we consider the rules of choosing points on an elliptic curve E . Let an elliptic curve $E : y^2 \equiv x^3 + ax + b \pmod{q}$ where $q \geq 5$. The number of points on

E is roughly estimated by letting $x = 0, 1, \dots, q-1$ and when $x^3 + ax + b$ is a square mod q . In order to determine the points on E , one needs to look at each possible $x \in F_q$ and then attempt to solve the equation for y . By Theorem 2.2 (Hasse), the number of points on an elliptic curve defined over a finite field F_q , must satisfy

$$|N - q - 1| < 2\sqrt{q} \quad (2.4)$$

N is the number of points on elliptic curve. It is important to understand the number of points on elliptic curves as it may help to understand the nature of the group, which is the important element in solving the discrete logarithm problem.

2.1.5 Discrete Logarithm Problem on Elliptic Curves

From the classical discrete logarithm problem: We know that $x \equiv g^k \pmod{q}$ for some k , and we need to find k . For elliptic curve version: Suppose we have points A, B on an elliptic curve E and we know that $B = kA (= A + A + \dots + A)$ for some integer k . We want to find k . This might not look like a logarithm problem, but it is clearly the analog of the classical discrete logarithm problem. Therefore, it is called the **discrete logarithm problem** for elliptic curves. For the attacks on discrete logarithm problem, see (Washington, 2003) and (Trappe & Washington, 2006).

2.1.6 Example on computing addition and doubling point on E

Let E be the curve $y^2 = x^3 + 7x + 12$ over field representation \mathbf{F}_{29} . To count points on E , we make a list of possible values of x then $x^3 + 7x + 12 \pmod{29}$, then of the square root y of $x^3 + 7x + 12 \pmod{29}$. This yields the points on E .

```
sage: E1=EllipticCurve(GF(29), [0,0,0,7,12])
```

```
sage: E1
```

```
Elliptic Curve defined by y^2 = x^3 + 7*x + 12 over Finite Field
```

of size 29

```
sage: show(plot(E1), aspect_ratio=1)
```

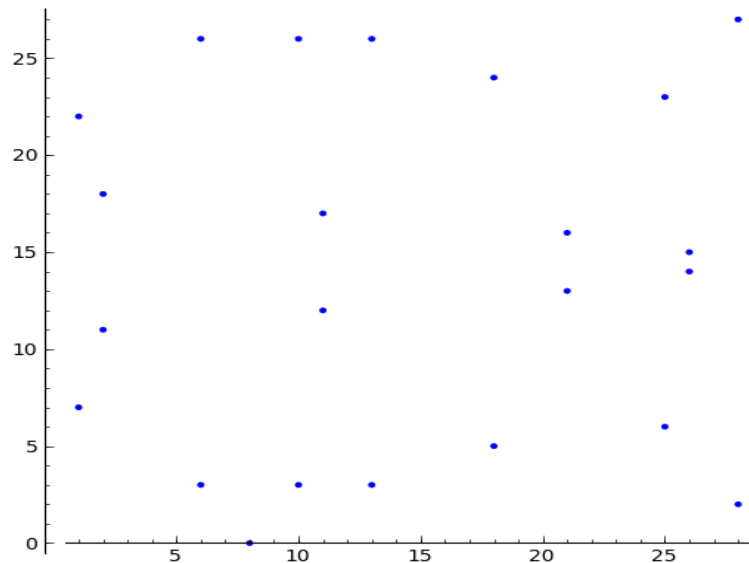


Figure 2.3: Elliptic curve $\mathbf{GF}(29)$

All the points are symmetric about the line $y = 14.5$. There are 24 points on E which is point ∞ and $(1,7)$, $(1,22)$, $(2,11)$, $(2,18)$, $(6,3)$, $(6,26)$, $(8,0)$, $(10,3)$, $(10,26)$, $(11,12)$, $(11,17)$, $(13,3)$, $(13,26)$, $(18,5)$, $(18,24)$, $(21,13)$, $(21,16)$, $(25,6)$, $(25,23)$, $(26,14)$, $(26,15)$, $(28,2)$, $(28,27)$

Therefore $E(\mathbf{F}_{29})$ has order 24.

By using Sage software, we randomly choose two different points for P and Q .

```
sage: P=E1.random_point()
sage: Q=E1.random_point()
sage: P,Q
((26 : 14 : 1), (6 : 3 : 1))
```

From the Sage software, we know that $P = (26, 14)$ and $Q = (6, 3)$. By referring section 2.1.3, compute the slope $m = \frac{y_Q - y_P}{x_Q - x_P}$

```
sage: F29=GF(29)
sage: F29((3-14)/(6-26))
2
```

Since the $x_P \neq x_Q$, we obtain the addition for P and Q using Matlab software:

```
>> addell([26, 14],[6,3],7, 12, 29)
ans =
     1     7
```

Therefore, $P + Q = R = (x_R, y_R) = (1, 7)$. And we can also obtain the double operation for P or Q as below:

```
>> multell([26, 14],2,7,12,29)
ans =
     2    11
>> multell([6,3],2,7,12,29)
ans =
    13     3
```

The result for both points are $2P = (2, 11)$ and $2Q = (13, 3)$.

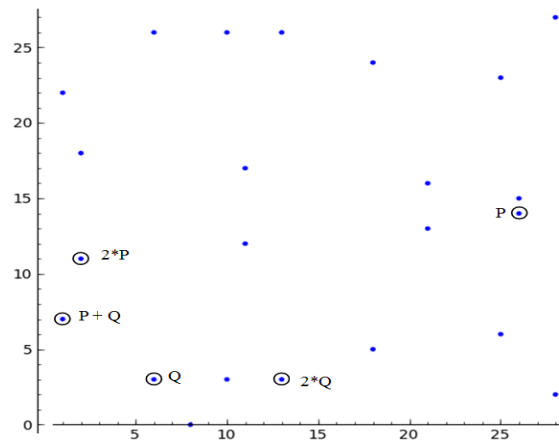


Figure 2.4: All the computation points on $\mathbf{GF}(29)$

2.2 Summary

This chapter presented the necessary mathematical background required for this thesis. The security in ECC relies on computing the mathematics using public-key created by multiplying two large primes over \mathbf{GF} . This makes the computational cost much higher compared to the modular exponentiation such as RSA (Rivest, Shamir & Adleman, 1978).

Therefore, we continue the study on how to pick suitable elliptic curves so that the order is trivial and the methods are vulnerable to provide equivalent result while using fewer bits in ECC. With the complete series of theories, we continue the next chapters by purposing algorithms for ECC in their key exchanges, encryption and digital signing documents.

CHAPTER 3

KEY AGREEMENT

Nowadays, the alarming advancement of technologies causes high requirement for secure communication. Due to the fact that algorithms using Asymmetric cryptosystems are much slower, most of the cryptosystems in the market are still based on Symmetric cryptosystems. But the communication may only work privately and limited only between authorized users. Therefore, we hope to improve more onto the Asymmetric cryptosystem especially on the study of key agreement scheme. Key agreement is a scheme that allows users to exchange and establish keys even under unsecured channel. It is schemes that are based on public-key which is initiate a conversation between two introduced users.

3.1 Introduction to Key Agreement

In the research of key agreement, we start the study with the most known key agreement scheme, the Diffie-Hellman Key Exchange (DHKE). The scheme was first published by Whitfield Diffie and Martin Hellman in 1976. It is a mathematical algorithm that allows two users to generate an identical shared secret key on both systems, even though those users may never have communicated with each other before. The scheme is based on the assumption that the discrete logarithm problem is intractable within a cyclic group, a passive adversary is not able to calculate or compute any information about the key, K . The basic steps for DHKE initiate a process between two users under the group of (\mathbf{G}, \bullet) . It is a group closed under the multiplication operation and $g \in \mathbf{G}$ is an element of the order n , both parameters (\mathbf{G}, \bullet) and g are published in public domain. The two users can be referred as Alice

and Bob, which is both of them chooses random integer numbers a and b within the interval $[0, n - 1]$ as the private-key. They create their public-key by computing $A = g^a$ and $B = g^b$ and exchange to each other to be multiply with their private-key. Alice obtains $K = B^a$ and Bob obtains $K = A^b$. From here, we know that both users have calculated same key using DHKE which is $K = g^{ba} = g^{ab}$. Further details of this scheme can be referred at Mel & Baker (2005) and Forouzan (2008).

We continue the study with the implementation of elliptic curves as the platform in generating the key exchange. It contains much more difficult set of problems for generating public and private keys, compare too many of the algorithms over the past decade. By using the combination discrete logarithm problem in ECC and DHKE, it is believed will make public key encryption more secure. The idea of the key exchange is known as Elliptic Curve Diffie Hellman (ECDH). For the basic steps in this protocol, we also represent between two users, Alice and Bob. They choose same elliptic curve E of the form $y^2 \equiv x^3 + ax + b \pmod{p}$ and a public base point $Q \in E$. Alice and Bob will choose a random integer a and b . It will be their private keys. Both of them are computing scalar multiplication to create their public key points, $A = aQ$ and $B = bQ$. Due to the discrete logarithm problem of elliptic curve, it is computationally infeasible to compute a and b even though A and B are public. Then, both users compute multiplication from each opponents' public key with their private keys to obtain $K = aB$ and $K = bA$. Any intruders will not able to compute the shared K as the secret keys a and b are not known. For more details for ECDH method can be referred at Washington (2003).

In 2005, a new key exchange method in cryptography has been introduced by Vladimir Shpilrain and Alexander Ushakov, it is known as Decomposition Problem (DP). The method is a study of exchanging keys between two different subgroups

by arranging certain parameters, depending on the particular group of G , which needs to be non-commutative group and denoted by the centralizer g . The method represent two ideas that improve the security of the key establishment which is, (a)The method conceal one of the subgroup A, B , and (b)The method make the user, Alice chooses her left private key as a_1 from one of the subgroup A, B and her right private key a_2 from the other subgroup. The same step goes to Bob. Here is the basic description of the protocol. Let say, the first user, Alice chooses an element in $a_1 \in G$ of length l , and a subgroup of $C_G(a_1)$, then she publishes its generators $A = \{\alpha_1, \dots, \alpha_k\}$. Bob also repeat the same, he chooses an element $b_2 \in G$ of length l , chooses a subgroup of $C_G(b_2)$, and publishes its generators $B = \{\beta_1, \dots, \beta_m\}$. Then Alice chooses a random element a_2 from $\langle \beta_1, \dots, \beta_m \rangle$ and sends the normal form $P_A = N(a_1 w a_2)$ to Bob. Similarly, Bob chooses a random element b_1 from $\langle \alpha_1, \dots, \alpha_k \rangle$ and sends the normal form $P_B = N(b_1 w b_2)$ to Alice. The key exchange can be obtain when Alice computes $K_A = a_1 P_B a_2$ and Bob computes $K_B = b_1 P_A b_2$. Since $a_1 b_1 = b_1 a_1$ and $a_2 b_2 = b_2 a_2$, so $K = K_A = K_B$, will be the shared key. Further details on this protocol can be referred at Shpilrain & Ushakov (2005).

Continue from the idea in DP, another new extension of key exchange primitive has been introduced by Yeşem Kurt (2006). This scheme known as Triple Decomposition Problem (TDP), relies on decomposing an element into three parts, where all are unknown. This seems to be harder problem since it requires quadratic systems to be solved instead of linear systems. The idea is to hide each of these components by multiplying them by random elements in subgroups. The crucial part is that one of the components is multiplied by random elements both on the right and on the left. The main difference of this scheme from Diffie-Hellman key

exchange is that in latter systems a known element is multiplied by elements on both sides whereas on this system an unknown element is multiplied by element on both sides. This ruins the linear relation between the public key and the private key. To find the private key or more accurately a key that works as a private key, an adversary has to decompose an element into three elements to satisfy certain conditions (Kurt, 2006). Here are the basic steps on describing the protocols. We have two users, Alice and Bob. Alice picks two elements $x_1, x_2 \in G$, chooses subsets S_{x_1} and S_{x_2} of $C(x_1)$ and $C(x_2)$ respectively, and publishes S_{x_1} and S_{x_2} . Bob also picks two elements $y_1, y_2 \in G$ chooses subsets S_{y_1} and S_{y_2} of $C(y_1)$ and $C(y_2)$ respectively, and publishes S_{y_1} and S_{y_2} . Then Alice chooses random elements $a_1 \in G, a_2 \in S_{y_1}$, and $a_3 \in S_{y_2}$. (a_1, a_2, a_3) is her private key. She sends Bob her public key (u, v, w) where $u = a_1x_1, v = x_1^{-1}a_2x_2, w = x_2^{-1}a_3$. Bob chooses random elements $b_1 \in S_{x_2}, b_2 \in S_{x_1}$, and $b_3 \in G$. (b_1, b_2, b_3) is his private key. He sends Alice his public key (p, q, r) where $p = b_1y_1, q = y_1^{-1}b_2y_2, w = y_2^{-1}b_3$. Therefore, Alice computes $a_1pa_2qa_3r = a_1b_1a_2b_2a_3b_3$ and Bob computes $ub_1vb_2wb_3 = a_1b_1a_2b_2a_3b_3$. For further details on the security and platform group G in this protocol, refer to (Kurt, 2006).

Since the idea of DP and TDP in cryptography still recent, we decided to continue the research by developing the same idea with implementing the elements from both methods using ECC. We want to find out that the concepts of developing key exchange ideas may also works in ECC because generating elements using points in ECC is easy to work with. But for unknown users, they will not able to compute the inverse within the reasonable amount of time because the security of ECC lies between solving the discrete logarithm problem within it (Mel & Baker, 2005). As the main research in this thesis, we will present two ideas of key exchange in ECC

by describing the algorithms within the related fields, and to strengthen the working method, we also included examples using mathematical programming Matlab and Sage.

3.2 Decomposition Problem in Elliptic Curve Cryptography

The operation of public key cryptographic scheme in Decomposition Problem in Elliptic Curve Cryptography (DPECC) involves arithmetic operation on an elliptic curve over a finite field determined by some domain parameter. In this section, we describe the elliptic curve parameters over finite field of order q denoted by F_q . ECC domain parameters over F_q are:

An indication field representation (FR) of the method used to representing field elements $\in F_q$, two field elements d and $e \in F_q$, that specifies the equation of the elliptic curve E over F_q (i.e., $y^2 = x^3 + dx + e$ for characteristic $q > 3$),

Assume that there will be two users involved in this key agreement; Alice and Bob, have no prior contact and the only communication channel between them is public. They both agreed on a same public point $Q \in [1, n - 1]$ on $E(F_q)$, a prime n is the order of Q , and the different cofactor picks from each users r or s where $r/s = \#E(F_q)/n$. $\#E(F_q)$ is the number of points on elliptic curve. And they will follow this sequence of steps for DPECC:

1. Alice chooses $a_1 = (x_{a_1}, y_{a_1})$ as her private key from group of G on $E(F_q)$. She picks $\#E(F_q)/n = r$ as the cofactor to generate her own subgroup. She uses a_1 to generates elements through the set. Then she gathers the points to be $A = \{\alpha_1, \alpha_2, \dots, \alpha_{r-1}\}$. She sends A to Bob.
2. Bob chooses $b_2 = (x_{b_2}, y_{b_2})$ as his private key from group of G on $\#E(F_q)$. He picks $\#E(F_q)/n = s$ as the cofactor to generate his own subgroup. He uses

b_2 to generate elements through the set. Then he gathers the points to be $B = \{\beta_1, \beta_2, \dots, \beta_{s-1}\}$. He sends B to Alice.

3. Alice gets subgroup of points B from Bob. From the list of $\langle \beta_1, \beta_2, \dots, \beta_{s-1} \rangle$, she picks one point to be her private key and defines it as a_2 . Then she multiplies it with public point Q and her first private key she chooses a_1 , to obtain

$$P_A = a_1 Q a_2 \quad (3.1)$$

Then she sends P_A to Bob.

4. Bob gets subgroup of points A from Alice. From the list of $\langle \alpha_1, \alpha_2, \dots, \alpha_{r-1} \rangle$, he picks one point to be his other private key and defined it as b_1 . Then he multiplies it with public point Q and his private key he chooses before b_2 , to obtain

$$P_B = b_1 Q b_2 \quad (3.2)$$

Then he sends P_B to Alice.

5. Alice obtains P_B from Bob, she multiplies it with her private key a_1, a_2 and defines it as:

$$K_A = a_1 P_B a_2 \quad (3.3)$$

6. Bob obtains P_A from Alice, he multiplies it with his private key b_1, b_2 and defines it as:

$$K_B = b_1 P_A b_2 \quad (3.4)$$

Since the curve $E(\mathbf{F}_q)$ is cyclic, so that we will have $a_1a_2 = a_2a_1$ and $b_1b_2 = b_2b_1$.

Their shared key will be defined from:

$$K_A = a_1P_Ba_2 = a_1b_1Qb_2a_2 \quad (3.5)$$

$$K_B = b_1P_Ab_2 = b_1a_1Qa_2b_2 \quad (3.6)$$

$\therefore K_A = K_B = K$, it shows that they shared the same key. Table 3.1 shows the summary of this method.

3.2.1 Summary of DPECC

Table 3.1: Summary of DPECC

Public: $E : y^2 = x^3 + dx + e, Q \in [1, n - 1]$	
Alice	Bob
<p>Chooses private key $a_1 = (x_{a_1}, y_{a_2})$</p> <p>She picks $r = \frac{\#E(F_q)}{n}$ as cofactor</p> <p>generates $A = \{\alpha_1, \alpha_2, \dots, \alpha_{r-1}\}$</p> <p>Sends $A \longrightarrow$</p>	<p>Chooses private key $b_2 = (x_{b_2}, y_{b_2})$</p> <p>He picks $s = \frac{\#E(F_q)}{n}$ as cofactor</p> <p>generates $B = \{\beta_1, \beta_2, \dots, \beta_{s-1}\}$</p> <p>$\longleftarrow$ Sends B</p>
<p>From subgroup of B, she picks</p> <p>$a_2 = \{\beta_1, \beta_2, \dots, \beta_{s-1}\}$ and</p> <p>compute $P_A = a_1 Q a_2$</p> <p>Sends $P_A \longrightarrow$</p>	<p>From subgroup of A, he picks</p> <p>$b_1 = \{\alpha_1, \alpha_2, \dots, \alpha_{r-1}\}$ and</p> <p>compute $P_B = b_1 Q b_2$</p> <p>\longleftarrow Sends P_B</p>
<p>Multiply P_B with a_1, a_2 to obtain</p> <p>$K_A = a_1 P_B a_2$</p>	<p>Multiply P_A with b_1, b_2 to obtain</p> <p>$K_B = b_1 P_A b_2$</p>
<p>Both obtain: $K = K_A = K_B = a_1 b_1 Q b_2 a_2 = b_1 a_1 Q a_2 b_2$</p>	

3.2.2 Example of DPECC

Defined the public domain:

q as the large prime number on $E : 101$

E as the elliptic curve: $y^2 = x^3 + 8x + 1$ with $d = 8, e = 1$

Generate E as follows: $E : y^2 \equiv x^3 + dx + e \pmod{101}$ where $d = 8$, and choose point $Q = (11, 39)$ to make sure point Q lie on E , then compute:

$$(39)^2 \equiv (11)^3 + (8)(11) + e \pmod{101}$$

$$e \equiv 102 \pmod{101}$$

$$\equiv 1 \pmod{101}$$

Q as the public point on E : $Q = (11, 39)$

n as an integer: $n = 97$

```
>> randprime(101)
```

```
ans =
```

```
97
```

Alice and Bob chooses r and s as their integer cofactor number where is

$$r/s = \frac{\#E(F_{101})}{97}$$

```
sage: E_101=EllipticCurve(GF(101), [0,0,0,8,1])
```

```
sage: E_101
```

```
Elliptic Curve defined by  $y^2 = x^3 + 8x + 1$  over Finite Field
```

```
of size 101
```

```
sage: show(plot(E_101), aspect_ratio=1)
```

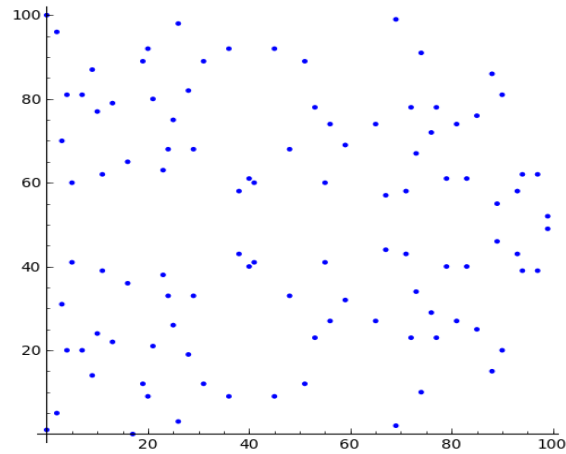


Figure 3.1: Elliptic Curve defined by $y^2 = x^3 + 8x + 1$ over Finite Field of size 101

1. Alice chooses $a_1 = (0, 1)$ as her private key, and random $r = 19$, then she generate subgroup of points from $19(0, 1)$

```
>> multsell([0,1], 19, 8, 1, 101)
```

2. She gathers the points as her subgroup called $A = \{(16, 36), (41, 60), \dots, (71, 43)\}$
She sends subgroup of A to Bob.

3. Bob chooses $b_2 = (4, 81)$ as his private key, and random $s = 59$, then he generate subgroup of points from $59(4, 81)$

```
>> multsell([4,81], 59, 8, 1, 101)
```

4. He gathers the points as his subgroup called $B = \{(3, 70), (4, 20), \dots, (99, 52)\}$
He sends subgroup of B to Alice.

5. From subgroup B , Alice chooses $a_2 = (73, 34)$. Then calculates:

$$P_A = a_1 Q a_2 = (0, 1) * (11, 39) * (73, 34)$$

```
>> addell([0,1], addell([11,39], [73,34], 8, 1, 101), 8, 1, 101)
```

```
ans =
```

```
41      60
```

$P_A = (41, 60)$. Then sends P_A to Bob.

6. From subgroup A , Bob chooses $b_1 = (88, 51)$. Then calculates:

$$P_B = b_1 Q b_2 = (88, 51) * (11, 39) * (4, 81)$$

```
>> addell([88,51], addell([11,39],[4,81], 8, 1, 101), 8, 1, 101)
```

```
ans =
```

```
13      22
```

$P_B = (13, 22)$. Then sends P_B to Alice.

7. Alice calculates $a_1 P_B a_2$

```
>> addell([0,1], addell([13,22],[73,34], 8, 1, 101), 8, 1, 101)
```

```
ans =
```

```
41      41
```

$$a_1 P_B a_2 = (41, 41) = K_A$$

8. Bob calculates $b_1 P_A b_2$

```
>> addell([88,15], addell([41, 60],[4,81], 8, 1, 101), 8, 1, 101)
```

```
ans =
```

```
41      41
```

$$b_1 P_A b_2 = (41, 41) = K_B$$

$$\therefore K = (41, 41) = K_A = K_B$$

3.3 Triple Decomposition Problem in Elliptic Curve Cryptography

Triple decomposition problem in ECC (TDPECC) is a protocol of decomposing into three elements for key exchange. We were adopting the idea from previous protocol by Yaşem Kurt (Kurt, 2006). As we have mentioned in section 3.2, the method generates certain subgroup, by picking random elements and multiplies it with three components under certain conditions. This method will break the linear relation between the public and private key (Kurt, 2006).

By referring the protocols in decomposition problem in elliptic curve (Zazali and Othman, 2009), we extend the ideas to the triple decomposition problem in elliptic curve (Othman and Zazali, 2009). Here, we consider the two users, Alice and Bob, both agreed on the same elliptic curve parameter over finite field of order q denoted by F_q . ECC domain parameters over F_q are: An indication FR of the method used to represent field elements $\in F_q$, two field elements d and $e \in F_q$ that specifies the equation of elliptic curve E over F_q (i.e., $y^2 = x^3 + dx + e$ for characteristic $p > 3$),

Assume that they agree on the same based point $Q = (x_Q, y_Q)$ on $E(F_q)$, and prime n which is the order of Q , and an integer $h = \#E(F_q)/n$ as the cofactor that generates list of points on the curve $E(F_q)$ by multiplying with Q . From this list of points, they do the following steps:

1. Alice picks her own points from the list, and creates a subset of points for herself known as $A = \langle A_1, A_2, A_3, X_1, X_2 \rangle$.
2. Bob picks his own points from the list, and creates a subset of points for himself known as $B = \langle B_1, B_2, B_3, Y_1, Y_2 \rangle$.
3. From the subset of points, Alice chooses random points to get $a_1 \in A_1, a_2 \in A_2, a_3 \in A_3, x_1 \in X_1, x_2 \in X_2$. Then she computes $u = a_1x_1, v = x_1^{-1}a_2x_2$,

$w = x_2^{-1}a_3$. She makes (a_1, a_2, a_3) as her private key and publishes (u, v, w) .

4. Bob does the same, from the subset of points, he chooses random points to get $b_1 \in B_1, b_2 \in B_2, b_3 \in B_3, y_1 \in Y_1, y_2 \in Y_2$. Then he computes $p = b_1y_1, q = y_1^{-1}b_2y_2, r = y_2^{-1}b_3$. He makes (b_1, b_2, b_3) as his private key and publishes (p, q, r) .

5. Alice gets (p, q, r) from Bob, and multiplies with her private key to obtain

$$a_1pa_2qa_3r = a_1(b_1y_1)a_2(y_1^{-1}b_2y_2)a_3(y_2^{-1}b_3) = a_1b_1a_2b_2a_3b_3 \quad (3.7)$$

6. Bob gets (u, v, w) from Alice and multiplies with his private key to obtain

$$ub_1vb_2wb_3 = (a_1x_1)b_1(x_1^{-1}a_2x_2)b_2(x_2^{-1}a_3)b_3 = a_1b_1a_2b_2a_3b_3 \quad (3.8)$$

Hence it shows that both agreed on the same shared key $K = a_1b_1a_2b_2a_3b_3$. Table 3.2 shows the summary of this method.

3.3.1 Summary of TDPECC

Table 3.2: Summary of TDPECC

Public: $Q \in E(F_q)$, cofactor $h = \#E(F_q)n$	
Alice	Bob
Creates her own subset of $A = \langle A_1, A_2, A_3, X_1, X_2 \rangle$	Creates his own subset of $B = \langle B_1, B_2, B_3, Y_1, Y_2 \rangle$
Chooses random points to get $a_1 \in A_1, a_2 \in A_2, a_3 \in A_3, x_1 \in X_1,$ $x_2 \in X_2$	Chooses random points to get $b_1 \in B_1, b_2 \in B_2, b_3 \in B_3, y_1 \in Y_1,$ $y_2 \in Y_2$
Computes: $u = a_1x_1, v = x_1^{-1}a_2x_2, w = x_2^{-1}a_3$	Computes: $p = b_1y_1, q = y_1^{-1}b_2y_2, r = y_2^{-1}b_3$
(a_1, a_2, a_3) as private key and publishes (u, v, w)	(b_1, b_2, b_3) as private key and publishes (p, q, r)
Gets (p, q, r) and computes $K_A = a_1pa_2qa_3r$	Gets (u, v, w) and computes $K_B = ub_1vb_2wb_3$
Both obtain: $K = K_A = K_B = a_1b_1a_2b_2a_3b_3$	

3.3.2 Example of TDPECC

Defined the public domain:

p as the large prime number on $E : 29$

E as the elliptic curve: $y^2 = x^3 + 24x + 13$ with $d = 24, e = 13$

Generate E as follows: $E : y^2 \equiv x^3 + dx + e \pmod{29}$ where $d = 24$, and choose point $Q = (1, 3)$ to make sure point Q lie on E , then compute:

$$(3)^2 \equiv (1)^3 + (24)(1) + e \pmod{29}$$

$$e \equiv -16 \pmod{29}$$

$$\equiv 13 \pmod{29}$$

Q as the public point on E : $Q = (1, 3)$

n as an integer: $n = 100$

Alice and Bob chooses r as their integer cofactor number where is $r =$

$$\frac{\#E(\mathbf{F}_{29})}{100}$$

```
sage: E_29=EllipticCurve(GF(29), [0,0,0,24,13])
```

```
sage: E_29
```

```
Elliptic Curve defined by y^2 = x^3 + 24*x + 13 over Finite Field  
of size 29
```

```
sage: show(plot(E_29), aspect_ratio=1)
```

1. Generate $100Q$ using Matlab to have points on finite field \mathbf{F}_{29} .

```
>> multsell([1,3],100,24,13,29)
```

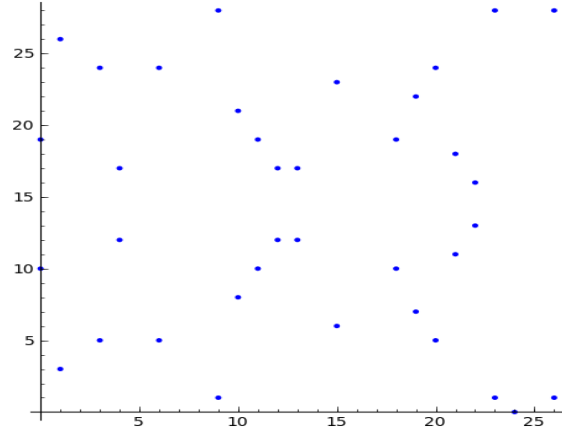


Figure 3.2: Elliptic Curve defined by $y^2 = x^3 + 24x + 13$ over Finite Field of size 29

2. Alice generates her own subset from the list of points on \mathbf{F}_{29} .

$$\langle A \rangle = (A_1, A_2, A_3, X_1, X_2)$$

$$A_1 = \{(11, 10), (23, 28), (20, 24)\}$$

$$A_2 = \{(4, 17), (15, 23), (18, 10)\}$$

$$A_3 = \{(15, 6), (0, 10), (4, 17)\}$$

$$X_1 = \{(19, 17), (1, 3), (0, 19)\}$$

$$X_2 = \{(\infty, \infty), (23, 1), (1, 26)\}$$

3. Bob generates his own subset from the list of points on \mathbf{F}_{29} .

$$\langle B \rangle = (B_1, B_2, B_3, Y_1, Y_2)$$

$$B_1 = \{(18, 10), (1, 26), (11, 10)\}$$

$$B_2 = \{(11, 19), (23, 28), (1, 3)\}$$

$$B_3 = \{(23, 1), (20, 24), (18, 19)\}$$

$$Y_1 = \{(11, 10), (19, 22), (\infty, \infty)\}$$

$$Y_2 = \{(15, 23), (19, 7), (4, 17)\}$$

4. Alice chooses $a_1 = (23, 28) \in A_1$, $a_2 = (4, 17) \in A_2$, $a_3 = (0, 10) \in A_3$, $x_1 =$

$$(1, 3) \in X_1, \quad x_2 = (1, 26) \in X_2$$

5. Bob chooses $b_1 = (11, 10) \in B_1$, $b_2 = (11, 19) \in B_2$, $b_3 = (18, 19) \in B_3$, $y_1 =$

$$(19, 22) \in Y_1, \quad y_2 = (15, 23) \in Y_2$$

6. Alice computes: $x_1 = (1, 3), x_1^{-1} = (1, 2)$ and $x_2 = (1, 26), x_2^{-1} = (1, 3)$

$$u = a_1 x_1 = (23, 28) * (1, 3)$$

```
>> addell([23,28], [1,3], 24,13,29)
```

```
ans =
```

```
0 10
```

$$v = x_1^{-1} a_2 x_2 = (1, 26) * (4, 17) * (1, 26)$$

```
>> addell([1,26], addell([4,17], [1,26], 24,13,29), 24,13,29)
```

```
ans =
```

```
20 24
```

$$w = x_2^{-1} a_3 = (1, 3) * (0, 10)$$

```
>> addell([1,3], [0,10], 24,13,29)
```

```
ans =
```

```
19 7
```

7. Bob computes: $y_1 = (19, 22), y_1^{-1} = (19, 17)$ and $y_2 = (15, 23), y_2^{-1} = (15, 6)$

$$p = b_1 y_1 = (11, 10) * (19, 22)$$

```
>> addell([11,10], [19,22], 24,13,29)
```

```
ans =
```

```
23 1
```

$$q = y_1^{-1} b_2 y_2 = (19, 7) * (11, 19) * (15, 23)$$

```
>> addell([19,7], addell([11,19], [15,23], 24,13,29), 24,13,29)
```

```
ans =
```

```
0 19
```

$$r = y_2^{-1}b_3 = (15, 6) * (18, 19)$$

```
>> addell([15,6], [18,19], 24, 13, 29)
```

```
ans =
```

```
18 10
```

8. Alice gets (p, q, r) and calculate:

$$a_1pa_2qa_3r = (23, 28) * (23, 1) * (4, 17) * (0, 19) * (0, 10) * (18, 10)$$

```
>> addell([23,28], addell([23,1], [4,17], 24, 13, 29), 24, 13, 29)
```

```
ans =
```

```
4 17
```

```
>> addell([4,17], addell([0,19], [0,10], 24, 13, 29), 24, 13, 29)
```

```
ans =
```

```
4 17
```

```
>> addell([4,17], [18,10], 24, 13, 29)
```

```
ans =
```

```
0 10
```

9. Bob gets (u, v, w) and calculate:

$$ub_1vb_2wb_3 = (0, 10) * (11, 10) * (20, 24) * (11, 19) * (19, 7) * (18, 19)$$

```
>> addell([0,10], addell([11,10], [20,24], 24, 13, 29), 24, 13, 29)
```

```
ans =
```

```
19 22
```

```
>> addell([19,22], addell([11,19], [19,7], 24, 13, 29), 24, 13, 29)
```

```
ans =
```

```
11 19
```

```
>> addell([11,19],[18,19],24,13,29)

ans =

     0     10
```

From the calculation using Matlab, we show that they obtain the same shared keys of $K = a_1pa_2qa_3r = ub_1vb_2wb_3 = (0, 10)$

3.4 Implementation issues

Our main goal in this research is to do some modification on the previous method in DP and TDP, so that it appears to be more secure, at least from the choices of the platform. Generally, we know that the security relies in ECC is based on solving the discrete logarithm problem. Therefore, we use ECC as the modification platform for DP and TDP.

The security in DPECC is apparently based on computing the cofactor number of points on the curve for r or s . It is computationally infeasible to obtain, if r or s large enough. Before arranging any attacks, the adversary would have to compute the number of points on elliptic curve by solving the discrete logarithm problem because $r/s = \#E(F_q)/n$.

As in TDPECC, the motivation of implementing the method in ECC is to reduce the problem of solving quadratic equations in generating elements for the protocol. In previous protocol, the system said to be at less capable from being altered by standard attack, but in this system, we can said that, it is very difficult to performing the attack since the basis of ECC is to solving the discrete logarithm problem for elliptic curves. The computation in this scheme seems to be longer because it employs steps of dividing keys into three different parts.

In the next chapters, we will present the idea of using the shared key, K (from

DPECC). Chapter 4, we will show how the computation of shared key K in the secured encryption method for ECC. And in Chapter 5, we will show how the key K could be implemented in signing digital messages. But, we do not discuss the shared key from TDPECC in the next chapters since certain implementation issues. We will continue the study of TDPECC in further research, thus it may also works in encryption and digital signing.

CHAPTER 4

ENCRYPTION IN CRYPTOGRAPHY

Encryption is the process of converting original information (plaintext) to another form of information (ciphertext), so that it will be readable to the authorized users only. In order to make the process useful, encryption needs keys to make it accessible. The basic idea in this process is the keys, which comes in pairs of an encryption and decryption keys. Encrypting messages can only be decrypted with the private-key. The private-key may only be known to the owner and must be kept secret, and it is infeasible to compute the private-key from public-key. Encryption algorithm determines how simple or complex the transformation process of the messages (or data). Encryption provides confidentiality, integrity and authentication of the information transferred between the users.

4.1 Introduction to Encryption

In this chapter, we focus on the study of asymmetric encryption schemes. We start with the encryption method introduced by Taher ElGamal in 1984. He was the first mathematician to propose asymmetric cryptosystem based on discrete logarithm problem. He proposed two distinct cryptosystems, one for encryption and the other for digital signatures (ElGamal, 1984), well before the elliptic curves were introduced in cryptography. The El-Gamal public-key encryption scheme used Diffie-Hellman key agreement protocol as the key transfer mode. The El-Gamal encryption is based on the same principle as Diffie-Hellman key exchange scheme which is a cyclic multiplicative group modulo some prime number. Here, we describe the El-Gamal encryption in ECC. The transaction is between two users, Alice and Bob who want

to communicate between each other over an insecure communication network. First, Bob chooses an elliptic curve E over a finite field \mathbf{F}_q , he chooses a point $Q \in E(\mathbf{F}_q)$, which usually the order is a large prime N . He chooses secret integer $s \in [1, N - 1]$ and computes $B = sQ$, another point on the curve. The public key becomes $(E(\mathbf{F}_q), N, Q, B)$. Alice, who has a message M which embeds in the point on E . For the encryption, Alice downloads Bob's public key, then she changes the messages into a point $M \in E(\mathbf{F}_q)$. Alice chooses a secret random integer k and computes $M_1 = kQ$. Then, she computes $M_2 = M + kB$ and sends pair of (M_1, M_2) to Bob. To decrypt the messages, Bob calculates $M = M_2 - sM_1$. This decryption works because $M_2 - sM_1 = (M + kB) - s(kQ) = M + k(sQ) - skQ = M$. We can see that Bob's secret integer s allows him to decrypt the message properly. Also, the assumption that is hard to compute in the discrete logarithm problem for elliptic curves gives an important responsibility. This is because, if there are any eavesdroppers who could solve the discrete logarithm problem, they could calculate s and therefore could retrieve the message M . For further details in this algorithm, refer ElGamal (1984).

We continue the study of encryption in El-Gamal by exploring a method of splitting messages introduced by Levent Ertaul and Weimin Lu in 2005. Split encryption is a method of splitting messages into n pieces before proceeding to using the El-Gamal encryption method. The messages can be split either before or after they send the messages to the receiver (Ertaul and Lu, 2005). They manage to use ECC as the platform to perform the split encryption. Suppose there will be a point Q on an elliptic curve $E(\mathbf{F}_q)$, with the order is a large prime N . Bob's private key is $n_B \in [0, N - 1]$ and compute the public key as $K_B = n_B Q$. The public key becomes $(E(\mathbf{F}_q), N, Q, K_B)$. For encryption, Alice converts the secret message, M

to a point P_M on the elliptic curve. Then, she uses El-Gamal encryption to get $P_1 = rQ$ and $P_2 = P_M + rK_B$. By letting $P_2 = (x_2, y_2)$, she chooses two random polynomials f_1, f_2 of degree $k - 1 \in E(\mathbf{F}_q)$ such that $f_1(0) = x_2, f_2(0) = y_2$, and split x_2, y_2 into n shares respectively. Then, Alice sends P_1 and n shares of $P_2(x_2, y_2)$ with their corresponding indices to Bob. To decrypt the messages, Bob recovers x_2 and y_2 and calculates the point $P_M = P_2 - n_B P_1$. This decryption works because $P_2 - n_B P_1 = (P_M + rK_B) - n_B(rQ) = P_M + r(n_B Q) - n_B r Q = M$. Then Bob converts the point P_M to the secret message, M . Instead of sending n pieces of x_2, y_2 to Bob, Alice chooses a random $k - 1$ degree polynomial f with $a_0 = x_2$ and $a_1 = y_2$. If they want to share more than one secret, they need to use Vandermonde matrix (Washington, 2003). According to the idea by Ertaul and Lu (2005), users may change the message M in two different way, (1) split before the encryption: Alice computes the n shares of secret message M first, then converts it to points $P_t \in E(\mathbf{F}_q)$, or (2) split after the encryption: Alice converts secret message M into a point P_t first, then she splits the point P_t into n shares of secret, let say we have $n = 3$ so that the messages split into $P_{t_1}, P_{t_2}, P_{t_3}$. For more details on this protocol, refer Ertaul & Lu (2005).

Before the El-Gamal, Massey-Omura scheme was introduced in 1983 by James Massey and Jim K. Omura, it is an encryption scheme based on three stages encryption protocol from the improvement of Shamir three-pass protocol (around 1980s). The Massey-Omura uses exponentiation in the Galois field for both encryption and decryption algorithm. It is schemes that can multipass protocol from key sharing, and it is suitable to prevent from man-in-the-middle attack. This method uses exponential in Galois field, $\mathbf{GF}(q)$ for both encryption and decryption. This is how the procedure be implemented in mathematical part. Let say we have, Alice and Bob

agree on an elliptic curve E over a finite field \mathbf{F}_q such that discrete logarithm problem is hard in $E(\mathbf{F}_q)$. Let $N = \#E(\mathbf{F}_q)$. For encryption Alice represents her message as a point $M \in E(\mathbf{F}_q)$. She chooses a secret integer m_A with $\gcd(m_A, N) = 1$, computes $M_1 = m_A M$, and sends M_1 to Bob. Then, Bob chooses a secret integer m_B with $\gcd(m_B, N) = 1$, computes $M_2 = m_B M_1$, and sends M_2 to Alice. To decrypt the message, Alice calculates $m_A^{-1} \in \mathbf{Z}_N$ and computes $M_3 = m_A^{-1} M_2$. Then she send M_3 to Bob. Bob calculates $m_B^{-1} \in \mathbf{Z}_N$ and computes $M_4 = m_B^{-1} M_3$. Therefore, he know that $M_4 = M$. This algorithm uses single elliptic curve as a platform, which is known. It foils the man-in-the-middle-attack, because it requires lot of communication. For further details how to convert message into points in elliptic curve, refer (Washington, 2003). For more details on this protocol, refer Massey & Omura (1983).

Here is the same idea as in El-Gamal which is the concept of splitting messages using Massey-Omura scheme by Ertaul & Lu (2005). Here is how it works. Let N be the order of $E_q(a, b)$ and we have two users, Alice and Bob. They choose secret point n_A and n_B respectively, such that $n_A \pmod{N}$ and $n_B \pmod{N}$. n_A^{-1}, n_B^{-1} is an integer representing the inverse of n_A, n_B . So we will have $n_A^{-1} n_A \equiv 1 \pmod{N}$ where $n_A^{-1} n_A = 1 + kN$ for some k , which also implies to n_B . The group $E(\mathbf{F}_q)$ has order N , referring the Lagrange's theorem, it implies that $NR = \infty$ for any point $R \in E(\mathbf{F}_q)$. For encryption, Alice wants to send message M to Bob. She splits the message into n shares of secret message M_t which is $1 \leq t < N$. She converts the secret message M_t into a point P_t as on the elliptic curve. Then create $P_{t_1} = n_A P_t$ and sends to Bob. Bob sends back $P_{t_2} = n_B P_{t_1}$ to Alice. To decrypt the message Alice sends back $P_{t_3} = (n_A^{-1} \pmod{N}) P_{t_2}$ to Bob. Bob decrypts the message using the inverse of his secret key, n_B^{-1} and obtain $(n_B^{-1} \pmod{N}) P_{t_3} = P_t$.

When Bob receives the k shares message from Alice, he finally gets the message P_M and converts P_M to the secret message M . The same concept goes to this method which is the messages can be split either before or after the user send the messages to the receiver. Since it is requires four transmissions between Alice and Bob, it have been found that it was not an efficient solution for threshold cryptosystems (Washington, 2003). Further details for this protocol also can be referred at Ertaul & Lu (2005).

With the study of El-Gamal, Massey-Omura and their split encryption in ECC, we continue the research by proposing an alternative method for encryption using the idea of decomposition keys in ECC (DPECC) from Chapter 3. This new algorithms are motivated from the techniques of designing encryption for ECC using El-Gamal and Massey Omura. Decomposition problem in Elliptic Curve Cryptography (DPECC) is a method of creating key exchange using two different subgroups, depending on the finite field group in $E(\mathbf{F}_q)$. By employing the hash function, the messages will be split into n shares before proceeding to the encryption. We only concentrate on the encryption study of DPECC in El-Gamal and Massey Omura because DPECC were developed from the use of two different secret/public keys. To strengthen the idea, there also an example included for the encryption concept in El-Gamal using the keys in DPECC.

4.2 El-Gamal encryption based on DPECC

Suppose that we have a point Q on an elliptic curve $E(\mathbf{F}_q)$ which usually the order is a large prime N .

From the method of generating keys in DPECC (Chapter 3), let Bob's private key as points of $b_1, b_2 \in \mathbf{F}_q$ and the public key is a combination of Alice's subgroup,

and public point of Q which is $K = K_B = b_1a_1Qa_2b_2$. The domain parameters in this method are $(E(\mathbf{F}_q), N, Q, K = K_B)$.

ENCRYPTION:

1. Alice converts the secret message M to a point P_M on the elliptic curve.
2. Alice chooses secret random integer r and using her private key a_1, a_2 , she computes $P_1 = ra_1Qa_2$ and $P_2 = P_M + rK_B$.
3. Let $P_2 = (x_2, y_2)$ by choosing two random polynomial f_1, f_2 of degree $k-1 \in \mathbf{F}_q$ such that $f_1(0) = x_2$ and $f_2(0) = y_2$ and split x_2, y_2 into n shares of secret message respectively. Alice sends P_1 and n shares of $P_2(x_2, y_2)$ with their corresponding to Bob.

DECRYPTION:

1. Bob recovers n shares of x_2, y_2 . Then he decrypt P_1 with his private key.
2. He retrieves the message with this step:

$$\begin{aligned}
 P_2 - b_1P_1b_2 &= (P_M + rK_B) - b_1(ra_1Qa_2)b_2 \\
 &= P_M + r(a_1b_1Qb_2a_2) - b_1(ra_1Qa_2)b_2 \\
 &= P_M
 \end{aligned}$$

3. Convert point P_M to get message, M .

From this method, we can say that, there will be two different Alice's and Bob's private key will be used to decrypt the message properly. Note that that the difficulty of obtaining Alice's and Bob's private key from the encrypted message is based on the discrete logarithm problem in ECC.

4.2.1 Example of El-Gamal encryption based on DPECC

This is an example of how Alice would send a message to Bob using the key exchange of DPECC that we calculate from previous chapter (section 3.2.2).

Bob chooses E to be $y^2 = x^3 + 8x + 1$ defined over \mathbf{F}_{101} and Q to be $(11, 39) \in E(\mathbf{F}_{101})$. Bob's public point $P_B = (13, 22)$, then he chooses $s = 59$ (using rand-prime.m) and calculates

$$sP_B = 59(13, 22) = (26, 98)$$

```
>> multell([13,22],59,8,1,101)
```

```
ans =
```

```
26    98
```

To send a message to Bob, Alice proceeds as follows.

1. Alice obtains Bob's public key and encodes her message as $P_M = (4, 20) \in E(\mathbf{F}_{101})$.

2. Alice chooses her secret integer $r = 19$ and computes

$$P_1 = ra_1Qa_2 = (26, 98)$$

```
>> multell([41,60],19,8,1,101)
```

```
ans =
```

```
26    98
```

$$\text{and } P_2 = P_M + a_1r(sP_B)a_2 = (4, 20) + (16, 65) = (13, 22)$$

```
>> adde11([4,20],[16,65],8,1,101)
```

```
ans =
```

```
13    22
```

3. Alice sends P_1 and P_2 to Bob.

4. Bob calculates $b_1(sP_1)b_2 = (3, 31)$

```
>> addell([88,15],addell([31,89],[4,81],8,1,101),8,1,101)
```

```
ans =
```

```
3    31
```

and $P_2 - b_1(sP_1)b_2 = (4, 20)$

```
>> addell([13,-22],[3,-31],8,1,101)
```

```
ans =
```

```
4    20
```

So Bob has securely received Alice's message $P_M = (4, 20)$.

4.3 Massey-Omura encryption based on DPECC

For this section, we want to show how the key exchange in DPECC may work on Massey-Omura encryption.

Let large prime N be the order of $E(\mathbf{F}_q)$, where a_1, a_2, b_1, b_2 is a secret points selected from Alice and Bob. There will be the inverse for each point so that $a_1a_1^{-1} \equiv 1 \pmod{N}$, $a_2a_2^{-1} \equiv 1 \pmod{N}$, $b_1b_1^{-1} \equiv 1 \pmod{N}$ and $b_2b_2^{-1} \equiv 1 \pmod{N}$. So the encryption is as follows:

ENCRYPTION:

1. Alice wants to send message M to Bob. She splits the messages into n shares of messages which is M_t where $1 < t < N$.
2. Alice converts the secret message, M_t into a point $P_t \in E(\mathbf{F}_q)$.

3. Alice calculates $a_1 P_t a_2 = P_{t_1}$. Then sends P_{t_1} to Bob.
4. Bob receives P_{t_1} and multiplies it with his private key and obtains $b_1 P_{t_1} b_2 = P_{t_2}$. Then he sends P_{t_2} back to Alice.

DECRYPTION:

1. Alice decrypts the message by multiplying with the inverse of her private key, $a_1^{-1} P_{t_2} a_2^{-1} = P_{t_3}$. She sends back P_{t_3} to Bob.
2. Bob decrypts P_{t_3} by multiplying it with the inverse of his private key and obtains $b_1^{-1} P_{t_3} b_2^{-1} = P_t$. He converts the point P_t to get message M .

As in the original Massey-Omura encryption, we show that this method also works using key exchange from DPECC. The working algorithm requires lots of communication and transaction to prevent from Man-In-The-Middle attack.

4.4 Implementation issues

Our goal in this chapter is to provide an encryption scheme, based on the idea of key exchange in DPECC from Chapter 3. We want to computationally prove that the idea could be done not only in key agreement but also for the encryption concept.

By referring the concept from El-Gamal and Massey-Omura encryption on ECC, we implement the keys from DPECC and split the messages into n pieces. The mathematical computation in both encryption seems to be easy to apply which is users choose large prime order of N , so that the choices of points are large enough. The purpose of this method is to make sure that the calculations for the inverse are much harder. It may prevent from Man-In-The-Middle attack, which is a type of attacker intrude into an existing connection to exchanged data and inject false information.

CHAPTER 5

DIGITAL SIGNATURE SCHEME

Digital signature scheme is way to ensure that an electronic document (e-mail, spreadsheet, text file, etc.) is authentic. Authentic means users know who created document and it has not been altered by any unauthorized parties. Digital signatures rely on certain types of encryption to ensure the authentication. Encryption is the process of taking all the data that one computer is sending to another and encoding it into a form that only the other computer will be able to decode. Authentication is the process of verifying that information is coming from a trusted source. These two processes work hand in hand for digital signatures.

5.1 Introduction to Digital Signature Algorithm

In this chapter, we start our study with the general view for digital signature scheme. In August 1991, the U.S National Institute of Standard and Technology (NIST) have proposed a digital signature scheme known as the Digital Signature Standard (DSS). They published DSS as FIPS 186, where uses a Digital Signature Algorithm (DSA) based on El-Gamal signature scheme. DSA is one of three signature scheme specified in FIPS 186 (Johnson and Menezes, 1999). To describe the protocol for DSA, let say we have two users, Alice and Bob, they need to defined the domain parameters where as select a 160-bit prime q and a 1024-bit prime p with the property that $q|p - 1$. Then, select a generator g of the unique cyclic group of order q in Z_p^* . An element of $h \in Z_p^*$ and compute $g \equiv h^{(p-1)/q} \pmod{p}$. Repeat until $g \neq 1$. So, the domain parameter are p, q and g . To generate the key pair, both users, select random or pseudorandom integer x such that $1 \leq x \leq q - 1$. Then compute $y \equiv g^x$

(mod p). Alice's public key is y and the private key is x . To sign a message m , Alice selects random integer k such that $1 \leq k \leq q - 1$. She computes $X \equiv g^k \pmod{p}$, $r \equiv X \pmod{q}$. Then computes $k^{-1} \pmod{q}$ and $e = SHA - 1(m)$. Then, she computes $s \equiv k^{-1}(e + xr) \pmod{q}$. Therefore, Alice's signature message is (m, r, s) . To verify the message, Bob obtains authentic copies of Alice's domain parameter (p, g, q) and public key y . He verify that r and s are integers in the interval $[1, q - 1]$. He computes $e = SHA - 1(m)$ and $w \equiv s^{-1} \pmod{q}$. Then he calculate $U_1 \equiv ew \pmod{q}, U_2 \equiv rw \pmod{q}$. to get $X \equiv g^{U_1}y^{U_2} \pmod{p}$ and $V \equiv X \pmod{q}$. Therefore, he verify the signature if $V = r$. The signature can be verified by computing V and then ratify if $r = V \pmod{q}$. For further details on this protocol, refer (Washington, 2003).

The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analogue of the DSA. ECDSA was first proposed in 1992 by Scott Vanstone in response to NIST's request for public comments on their first proposal to DSS. It was accepted in 1998 as an International Standard Organization standard (ISO 14888-3), accepted in 1999 as an ANSI (American National Standard Institute) standard (ANSI X9.62), and accepted in 2000 as an IEEE (Institute of Electrical and Electronic Engineers) standard (IEEE 1363-2000) and FIPS standard (FIPS 186-2). It is also under consideration for inclusion in some other ISO standard (Johnson and Menezes, 1999). The ECDSA schemes are generally describe as a transaction between two users, Alice and Bob. Let say, Alice wants to sign a message m , which is integer. She chooses a prime p and an elliptic curve $E \pmod{p}$. The number of points n on E is computed and a large prime factor q of n is found. A point $A(\neq \infty)$ is chosen such that $qA = \infty$. The message m must satisfy $0 \leq m < q$. Alice generate pair of key by associated with a particular set of EC domain parameters (E, q, A, n) .

Alice chooses her secret integer a and computes $B = aA$. The public information is p, E, q, A, B and a kept as secret. Alice signs the message m using EC domain parameters (E, q, A, n) and key pairs of (B, a) . She chooses a random integer k with $1 \leq k < q$ and computes $R = kA = (x, y)$. Then, she computes $e = SHA-1(m)$ and $s \equiv k^{-1}(e + aR) \pmod{q}$. She sends the signed message (m, R, s) to Bob. To verify the signing messages, firstly Bob needs to obtain an authentic copy of Alice's domain parameters (E, q, A, n) and associated with public key B . Then Bob verifies the signature by computes $e = SHA-1(m)$. Then, he computes $u_1 \equiv s^{-1}e \pmod{q}$ and $u_2 \equiv s^{-1}R \pmod{q}$ and $V = u_1A + u_2B$. If $V = \mathcal{O}$, otherwise accept V . He declares the signature valid if $V = R$. Conceptually, the ECDSA is simply obtained from the DSA algorithm, refer Johnson and Menezes (2005) for further details on the comparison between the protocols.

We have study how the implementation of decomposition problem in ECC for key agreement in Chapter 3 and encryption in Chapter 4. Therefore, in the next section, we will study how DPECC may works for signature scheme in verifying and signing electronic messages.

5.2 Decomposition Problem in Elliptic Curve Cryptography in Digital Signature Algorithm

To develop an algorithm based on DSA, it must consist of four steps for signing the electronic messages, such as:

1. A *domain parameter generation algorithm* that generates a set D of domain parameters.
2. A *key generation algorithm* that takes as input a set D of domain parameters and generates key pairs (K_A, P_B, a_1, a_2) .

3. A *signature generation algorithm* that takes as input a set of domain parameters D , a private key a_1, a_2 , and a message m , and produces a signature Σ .
4. A *signature verification algorithm* that takes as input the domain parameters D , a public key K_A, P_B , a message m , and a purported signature Σ , and accepts or rejects the signature.

We assume that the domain parameter D are valid, with the key generation using DPECC and SHA-1 denotes a cryptographic hash function whose output have bitlength not more than N , the signature scheme follows as below:

DOMAIN PARAMETERS: Alice and Bob agreed on the same public domain on elliptic curve field such as:

1. Two field elements a and $b \in \mathbf{F}_q$, as the elements in elliptic curve E over \mathbf{F}_q (i.e., $y^2 = x^3 + ax + b$ for characteristic $q > 3$).
2. Both agreed on the same public point $Q \in E(\mathbf{F}_q)$ with the order of prime q .
3. Domain parameter $D = (E(\mathbf{F}_q), Q, q, N)$.

DPECC KEY PAIR GENERATION: For the key generation, Alice will follow these steps based on key generation in DPECC method:

1. Alice chooses her own private keys a_1 and a_2 .
2. Alice gets P_B which is the public key generate by Bob.
3. She multiplies P_B with her own private key a_1 and a_2 to obtain K_A .
4. The public information is $E(\mathbf{F}_q), Q, N, q, P_B$ and K_A .

DPECC SIGNATURE GENERATION: Using the domain parameter $D = (E(\mathbf{F}_q), Q, N, q, P_B, K_A)$, private key a_1, a_2 and message m , Alice generates the digital signature as below:

1. Select a random or pseudorandom integer k which is $1 \leq k \leq q - 1$.
2. Computes $kP_B = (x_1, y_1)$ and $R \equiv x_1 \pmod{q}$. If $R = 0$, go step 1.
3. Computes $k^{-1} \pmod{q}$.
4. Computes $e = SHA - 1(m)$.
5. Computes $s \equiv k^{-1}(e + a_1a_2R) \pmod{q}$. If $s = 0$, go step 1.
6. Alice's signature message is (m, R, s) and she sends it to Bob.

DPECC VERIFICATION: Bob obtains an authentic copy of Alice's domain parameters $D = (E(\mathbf{F}_q), Q, N, q, P_B, K_A)$, public key K_A, P_B , message m and signature (m, R, s) .

1. Verify that R and s are integers in the interval $[1, q - 1]$. If any verification fails, then reject the signature.
2. Computes $e = SHA - 1(m)$.
3. Computes $U_1 \equiv es^{-1} \pmod{q}$ and $U_2 \equiv Rs^{-1} \pmod{q}$
4. Computes $V = U_1P_B + U_2K_A$. If $V = \mathcal{O}$, then reject the signature. Otherwise compute.
5. Convert the x -coordinate x_1 of R and computes $x_1 \pmod{q}$.
6. If $V = R$, then accept the signature. Else, reject the signature.

Proof:

$$V = (U_1 + U_2 a_1 a_2) P_B$$

$$\text{Let } s = k^{-1}(e + a_1 a_2 R)$$

$$k \equiv s^{-1}(e + a_1 a_2 R)$$

$$\equiv s^{-1}e + s^{-1}a_1 a_2 R$$

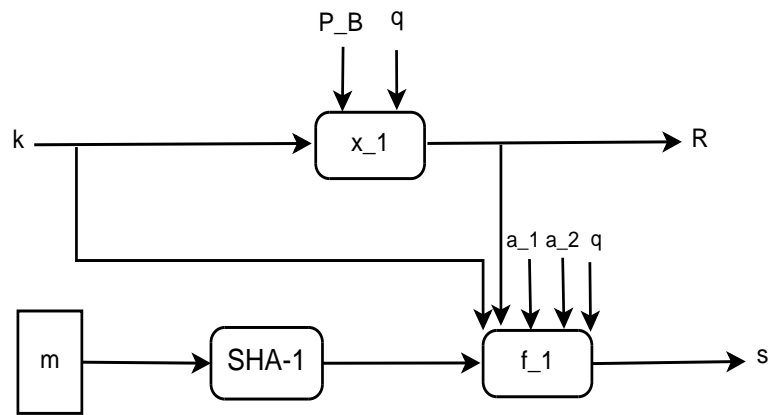
$$\equiv (U_1 + U_2 a_1 a_2) \pmod{q}$$

so the signature can be verified by computing V and then ratify if $V = R$ as required.

5.3 Summary of DPECC in Signing Digital Messages

Table 5.1: Summary of DPECC in Signing and Verifying Digital Messages

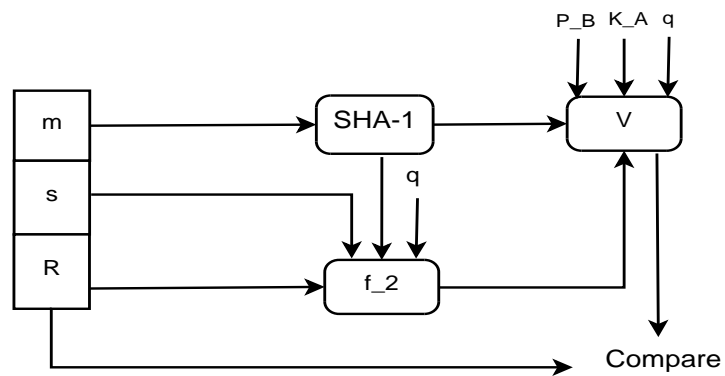
<u>Global public-key component:</u> Q : public point with $[1, n - 1]$ \mathbf{F}_q : finite field of order q $\#E(\mathbf{F}_q)$: number of points on E	
<u>Alice's private key</u> a_1 : random or pseudorandom point from $\#E(\mathbf{F}_q)$ a_2 : random or pseudorandom point from $\beta_1, \dots, \beta_{s-1}$	<u>Alice's public key</u> K_A : key exchange generate by Alice (Refer Section 3.2: $K_A = K_B = K$) P_B : public key from Bob
<u>Alice's per-message secret number:</u> k : random or pseudorandom point from $1 \leq k \leq q - 1$	
<u>Signing</u> $k P_B = (x_1, y_1)$ and $R \equiv x_1 \pmod{q}$ $s \equiv k^{-1}(\text{SHA-1}(m) + a_1 a_2 R) \pmod{q}$ Signature: (m, R, s)	<u>Verifying</u> $U_1 \equiv \text{SHA-1}(m) s^{-1} \pmod{q}$ $U_2 \equiv R s^{-1} \pmod{q}$ $V = U_1 P_B + U_2 K_A$ Test: if $V = R$; accept



$$s = f_1(\text{SHA-1}(m), a_1, a_2, R, q) = (k^{-1})(e + a_1 a_2 R) \pmod{q}$$

$$R = (k, P_B, x_1, q) = x_1 \pmod{q}$$

Figure 5.1: DPECC Signing



$$U_1 = f_2(s, \text{SHA-1}(m), q) = e s^{-1} \pmod{q}$$

$$U_2 = f_2(s, q, R) = R s^{-1} \pmod{q}$$

$$V = (U_1, P_B, U_2, K_A, q) = U_1 P_B + U_2 K_A$$

Figure 5.2: DPECC Verifying

5.4 Implementation Issues

By completing this chapter, we conclude that Decomposition Problem from the original work by Shpilrain & Ushakov could be implemented in finite field of elliptic curves arithmetic. The intractability of EC Discrete Logarithm Problem (ECDLP) in ECC give a higher complexity in computing ECDLP even with uses of smaller parameters. Even with smaller parameters, it give better level of security to help in faster computations, utilization of less power consumption, bandwidth, storage space and other constrained resources such as processing power. The signature scheme based on elliptic curves can be used for data authentication, data integrity and non-repudiation (Hankerson et al., 2003).

The key generated by the implementation of DPECC is secured and it consumes smaller key size used by the elliptic curves. From the algorithm, it shows that the users can verify the messages using DPECC, the working part seems to be longer since we split the messages using $SHA - 1$ function. To achieve an outstanding result, we consider the protocol needs to include certain particular consideration such as the suitability of elliptic curve arithmetic (in point addition, point doubling, point multiplication) using DPECC.

CHAPTER 6

CONCLUSION

In this study, we use the elliptic curves as algebra in all over the mathematics discussed. The ability of elliptic curves is to take any two points on a specific curve, add them together, and get another point on the same curve. More importantly for cryptography, is the difficulty of figuring out which two points are added together to get the answer. For the right choice of various parameters, that difficulty is exponential with key length. The cryptanalyst must use a very advanced mathematics even in the early attempt to crack a code where it does not take many bits because the task is practically impossible.

We started the study with the basis of several Asymmetric cryptosystem in key management cryptography, and one of the study that attracted the interest is in the idea of Decomposition Problem (Shpilrain and Ushakov, 2005) and Triple Decomposition method (Kurt, 2006). Both ideas are based on developing keys on non-commutative (infinite) groups and we manage to develop the same idea in ECC commutative (finite) groups. We called both method as Decomposition Problem in Elliptic Curve Cryptography and Triple Decomposition Problem in Elliptic Curve Cryptography. The rational of this implementation seem to be practical without infringing the main concept of the protocols.

We continue the study in encryption for ECC because we want to investigate the effectiveness of the key generation that has been done in DPECC. We were motivated from the ideas of splitting encryption scheme from El-Gamal (1985) and Massey-Omura (1983). Modifications are made on this algorithm by using keys from DPECC. From this encryption study, we did not continue the encrypted algo-

rithm using TDPECC because we find out that the scheme give us longer encrypted algorithm compare to DPECC key exchange.

We complete the research by designing digital signature in ECC. We design the digital signature using key exchange in DPECC by adopting the idea of former digital signature from Elliptic Curve Digital Signature Algorithm (ECDSA).

But in this thesis, we did not continue the scheme using key exchange in TDPECC for encryption and digital signature. One of the reasons is the scheme take us longer time to complete the rotation of the algorithm. But we are positively ensured that we can complete this part of research in our future works.

Appendix A

Elliptic Curves

A.1 Weierstrass equation

Elliptic curves are not ellipses. It was from the relation of elliptic integrals such that

$$\int_{z_1}^{z_2} \frac{dx}{\sqrt{x^3 + bx + c}} \quad \text{and} \quad \int_{z_1}^{z_2} \frac{xdx}{\sqrt{x^3 + bx + c}}$$

that arises between the computations of the arc length of ellipses.

Elliptic curves are interesting studies in cryptography. We can use any two points on the curve to produce a third point on the curve.

The set of points (x, y) on elliptic curve E is defined from the **Weierstrass equation**:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1.1}$$

where $a_1, a_2, a_3, a_4, a_6 \in \mathbf{F}$ and $\Delta \neq 0$, where Δ is the discriminant of E and defined as follows:

$$\Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6$$

$$d_2 = a_1^2 + 4a_2$$

$$d_4 = 2a_4 + a_1a_3$$

$$d_6 = a_3^2 + 4a_6$$

$$d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2$$

Elliptic curve E is defined over \mathbf{F} because the coefficients a_1, a_2, a_3, a_4, a_6 are elements of \mathbf{F} , where \mathbf{F} is called the underlying field. The condition $\Delta \neq 0$ ensures

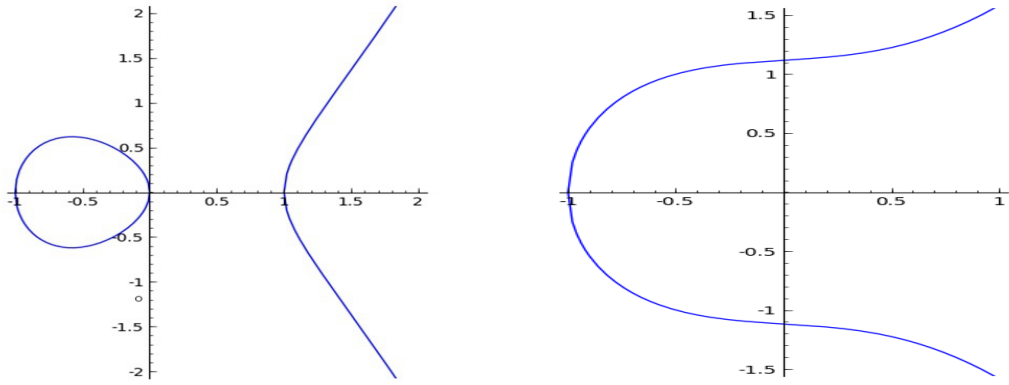


Figure A.1: Elliptic Curves examples on \mathbf{R}

that elliptic curve E is smooth, which is there are no points at which curve has two or more distinct tangent lines.

Let \mathbf{F} be the real numbers with the characteristic q larger than 3. Set of points where ∞ denotes the point at infinity:

$$E(\mathbf{F}) = \{(x, y) \in \mathbf{F} \times \mathbf{F}; y^2 = x^3 + Ax + B\} \cup \{\infty\}$$

If \mathbf{F} is a field with $A, B \in \mathbf{F}$, then we can say that the elliptic curve E is defined over \mathbf{F} . In general, we use E and \mathbf{F} to represent the field and write it as $E(\mathbf{F})$. Point of infinity, $\{\infty\}$ are included on elliptic curves for the use in the group operation defined in following section.

A.1.1 Simplified Weierstrass equation

A Weierstrass equation in equation

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

defined over \mathbf{F} can be simplified depending on the characteristic q of field \mathbf{F} . Consider the separate cases as below:

1. If the characteristic of field \mathbf{F} defined over prime $q \notin \{2, 3\}$ with the field \mathbf{F}_q

transforms elliptic curve E to the curve

$$y^2 = x^3 + Ax + B \quad (1.2)$$

where $A, B \in \mathbf{F}$. The discriminant of the curve is $\Delta = -16(4A^3 + 27B^2) \neq 0$

2. If the characteristic of field \mathbf{F} defined over prime $q = 2$ with the field \mathbf{F}_{2^m} , then there are two cases to consider.

- If $a_1 \neq 0$, elliptic curve E transforms to the curve

$$y^2 + xy = x^3 + Ax^2 + B \quad (1.3)$$

where $A, B \in \mathbf{F}$. And the curve is said to be non-supersingular with discriminant $\Delta = B \neq 0$.

- If $a_1 = 0$, elliptic curve E transforms to the curve

$$y^2 + Cy = x^3 + Ax + B \quad (1.4)$$

where $A, B, C \in \mathbf{F}$. And the curve is said to be supersingular with discriminant $\Delta = C^4 \neq 0$.

3. If the characteristic of \mathbf{F} defined over prime $q = 3$ with the field \mathbf{F}_{3^m} , then there are two cases to consider.

- If $a_1^2 \neq -a_2$, elliptic curve E transforms to the curve

$$y^2 = x^3 + Ax^2 + B \quad (1.5)$$

where $A, B \in \mathbf{F}$. And the curve is said to be non-supersingular with discriminant $\Delta = -A^3B \neq 0$.

- If $a_1^2 = -a_2$, elliptic curve E transforms to the curve

$$y^2 = x^3 + Ax^2 + B \quad (1.6)$$

where $A, B \in \mathbf{F}$. And the curve is said to be supersingular with discriminant $\Delta = -A^3 \neq 0$.

A.2 Group Law

In this section, we describe how the computation works from two points on an elliptic curve (or even one) to produce another point. Let E be an elliptic curve defined over the field \mathbf{F} . There is a *chord-and-tangent rule* for adding two points in $E(\mathbf{F})$ to give third point in $E(\mathbf{F})$. The set of points $E(\mathbf{F})$ forms an abelian group with ∞ as its identity.

The addition rule can be explained by let $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be the two distinct points on an elliptic curve E . The sum R defined by draw a line through P and Q , and this line intersects the elliptic curve at third point. Then R is the reflection of this point about the x -axis. This can be shown in Figure 2.2.

The double rule of R defined by draw a tangent line to the elliptic curve at point P . The line intersects the elliptic curve at a second point. Then R is the reflection of this point about the x -axis.

Algebraic formulas for the group law can be derived from the geometric description. In this thesis, we presented the elliptic curve E of the simplified Weierstrass form 1.2 in affine coordinates when the characteristic of the underlying field \mathbf{F} is not 2 or 3 (e.g., \mathbf{F}_q where $q > 3$ is a prime).

A.2.1 Addition and Doubling operation

As a summary, from an elliptic curve E , defined by $y^2 = x^3 + ax + b$ over a finite field \mathbf{F}_q , with characteristic larger than 3. Let $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be the points on E with $P, Q \neq \infty$. Then define that $P + Q = R = (x_R, y_R)$ as follows:

1. (Point addition) If $P \neq Q$, then

$$x_R = m^2 - x_P - x_Q$$

$$y_R = m(x_P - x_R) - y_P$$

$$\text{where } m = \frac{y_Q - y_P}{x_Q - x_P}$$

2. (Point doubling) If $P = Q$, and $y_P \neq 0$, then

$$x_R = m^2 - 2x_P$$

$$y_R = m(x_P - x_R) - y_P$$

$$\text{where } m = \frac{3x_P^2 + A}{2y_P}$$

3. (Identity) If $P = Q$ and $y_P = 0$, then $P + Q = \infty$. Also define that $P + \infty = P$ for all points P on E .

4. (Negatives) $P = Q$, but $y_P \neq y_Q$, then $P + Q = \infty$.

For the characteristic of \mathbf{F}_q is 2 or 3, then we use the same method for elliptic curve addition but the formula are different.

A.3 Group order

Let E be an elliptic curve defined over \mathbf{F}_q . The number of points in $E(\mathbf{F}_q)$, denoted $\#E(\mathbf{F}_q)$, is called the *order* of E over (\mathbf{F}_q) . Since the Weierstrass equation (1.1) has at most two solutions for each $x \in (\mathbf{F}_q)$, we know that $\#E(\mathbf{F}_q) \in [1, 2q + 1]$. Hasse's theorem provides tighter bounds for $\#E(\mathbf{F}_q)$.

Here we will proof how the Hasse theorem in Theorem 2.2 works.

Proof. Let

$$a = q + 1 - \#E(\mathbf{F}_q) = q + 1 - \deg(\phi_q - 1)$$

We want to show that $|a| \leq 2\sqrt{q}$. We need the following

Lemma A.1. *Let r, s be integers with $\gcd(s, q) = 1$. Then $\deg(r\phi_q - s) = r^2q + s^2 - rsa$*

Since $\deg(r\phi_q - s) \geq 0$, the lemma implies that

$$q\left(\frac{r}{s}\right)^2 - a\left(\frac{r}{s}\right) + 1 \geq 0$$

for all r, s with $\gcd(s, q) = 1$. The set of rational numbers r/s such that $\gcd(s, q) = 1$ is dense in \mathbf{R} . (*Proof:* Take s to be a power of 2 or a power of 3, one of which must be relatively prime with q . The rationals of the form $r/2^m$ and those of the form $r/3^m$ are easily seen to be dense in \mathbf{R} .) Therefore,

$$qx^2 - ax + 1 \geq 0$$

for all real number x . Therefore the discriminant of the polynomial is negative or 0, which means that $a^2 - 4q \leq 0$, hence $|a| \leq 2\sqrt{q}$.

□

A.4 Number theory

- The *greatest common divisor* (\gcd), of two non-zero integers, is the largest positive integer that divides both numbers.
- The integers a and b are said to be *coprime* if they have no common factor other than 1 or -1 , or equivalently, if their \gcd is 1.
- The *Euler totient function* $\phi(n)$ of positive integer n is defined to be the number of positive integers less than or equal to n and coprime to n . For example, $\phi(8) = 4$ since the four numbers 1, 3, 5 and 7 are coprime to 8, but 2, 4 and 6 are not.

- Let n be a positive integer. Then \mathbf{Z}_n is the set of integers modulo n :

$$\mathbf{Z}_n = \{0, 1, 2, \dots, n - 1\}$$

and \mathbf{Z}_n is a group under addition. Define \mathbf{Z}_n^* as

$$\mathbf{Z}_n^* = \{a \mid 1 \leq a \leq n, \gcd(a, n) = 1\}$$

\mathbf{Z}_n^* is a group with respect to multiplication mod n .

- Let $a \in \mathbf{Z}_n^*$. The *order of a mod n* is the smallest integer $k > 0$ such that $a^k \equiv 1 \pmod{n}$. The order of a mod n divides $\phi(n)$ (the Euler totient function).
- A *primitive root modulo n* is an integer g such that, modulo n , every integer coprime to n is congruent to a power of g . Consider, for example, when $n = 14$ so $\mathbf{Z}_n^* = \{1, 3, 5, 9, 11, 13\}$. We see that 3 is a primitive root modulo 14 as

$$\{3^1, 3^2, 3^3, 3^4, 3^5, 3^6\} = \{3, 9, 27, 81, 243, 729\} \equiv \{3, 9, 13, 11, 5, 1\} = \mathbf{Z}_n^*$$

The only other primitive root modulo 14 is 5.

- Let p be prime and $a \in \mathbf{Z}_p^*$. The order of a mod p divides $(p - 1)$. A *primitive root mod p* is an integer, g , such that the order of g mod p equals $(p - 1)$. Then every integer is congruent modulo p to 0 or a power of g . For example, 3 is a primitive root mod 7:

$$\{1, 3, 9, 27, 81, 243\} \equiv \{1, 3, 2, 6, 4, 5\} \pmod{7} \equiv \mathbf{Z}_7^*$$

There are $\phi(n - 1)$ primitive roots mod p . A primitive root mod p always exists and so \mathbf{Z}_p^* is a cyclic group.

Theorem A.1. (Chinese Remainder theorem) *Let n_1, n_2, \dots, n_r be positive integers such that $\gcd(n_i, n_j) = 1$ when $i \neq j$. Let a_1, a_2, \dots, a_r be integers. There exist an x such that*

$$x \equiv a_i \pmod{n_i} \text{ for all } i$$

The integer x is uniquely determined modulo $n_1 n_2 \dots n_r$.

Example A.1. Let $n_1 = 4, n_2 = 3, n_3 = 5$ and let $a_1 = 1, a_2 = 2, a_3 = 3$. Then $x = 53$ is a solution to the simultaneous congruences

$$x \equiv 1 \pmod{4}, \quad x \equiv 2 \pmod{3}, \quad x \equiv 3 \pmod{5}$$

and any solution to the congruences is equivalent to 53 modulo 60.

Theorem A.2. (Fermat's little theorem). *If p is a prime number then for any integer a*

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

A.5 Group Theory

- A *set* is a collection of objects considered as a whole. The objects of a set are called *elements*. If A and B are sets and every element of A is also an element of B , then A is a *subset* of B .
- A *group* $(G, *)$ is a nonempty set, G together with a group operator, $*$, which satisfy the group axioms:

- *Associativity:* $\forall a, b, c \in G, \quad (a * b) * c = a * (b * c)$

- *Identity element:* $\exists e \in G$ such that $\forall a \in G, \quad e * a = a * e = a$

- *Inverse element:* $\forall a \in G \exists b \in G$ such that, $a * b = b * a = e$

(where e is the neutral element).

- *Closure*: $\forall a, b \in G \quad a * b \in G$

- A group G is said to be *abelian* (or commutative) if for every $a, b \in G$, $a * b = b * a$. Groups lacking this property are called *non-abelian*.
- The integers under addition form an abelian group while the integers under multiplication do not (as not every integer has an inverse that is also an integer under multiplication)
- If the operation is thought of as an analogue of multiplication, then the group operations are written multiplicatively. That is:
 - write $a \cdot b$ or even ab for $a * b$ and call it the product of a and b .
 - write 1 (or e) for the identity element and call it the unit element.
 - write a^{-1} for the inverse of a and call it the reciprocal of a .

However, sometimes the group operation is thought of as analogous to addition and written additively:

- write $a + b$ for $a * b$ and call it the sum of a and b .
- write 0 for the identity element and call it the zero element.
- write $-a$ for the inverse of a and call it the opposite of a .

Usually, only abelian groups are written additively, although abelian groups may also be written multiplicatively.

- As elliptic curves form additive abelian groups we use additive group notation in this project (although we use ∞ for the identity element)
- The *order* of a group G , denoted by $|G|$, is the number of elements of the set G . A group is called *finite* if it has finitely many elements.

- The *order of an element* $g \in G$ is the smallest integer $k > 0$ such that $g * g * \dots * g$ (k times) $= e$. So using the additive notation of this product the order of $g \in G$ is the smallest integer $k > 0$ such that $kg = 0$. Note that if k is the order of g then

$$g^i = g^j \Leftrightarrow i \equiv j \pmod{k}$$

- Given a group G under binary operation $*$, we say that a subset H of G is a *subgroup* of G if H also forms a group under the operation $*$.

Theorem A.3. (Lagrange's theorem). *Let G be a finite group.*

- Let H be a subgroup of G . Then the order of H divides the order of G .*
- Let $g \in G$. Then the order of g divides the order of G .*

Consider two sets of elements, the domain and the codomain, and a function f that maps elements from the domain to the codomain.

- f is *injective* (1 – 1) if, for every y in the codomain, there is at most one x in the domain such that $f(x) = y$.
- f is *surjective* (onto) if, for every y in the codomain, there is at least one x in the domain such that $f(x) = y$.
- f is *bijective* if, for every y in the codomain, there is exactly one x in the domain such that $f(x) = y$.

So the function f is bijective if it is both injective and surjective.

- A *homomorphism* is a structure-preserving map between two algebraic structures (such as groups, rings, or vector spaces). So a homomorphism between groups preserves the structure of the group operation.

- An *isomorphism* is a bijective (1 – 1 & onto) map f such that both f and its inverse f^{-1} are homomorphisms.
- An *automorphism* is an isomorphism from an object to itself
- An *endomorphism* is a homomorphism from an object to itself.

The diagram below denotes implication.

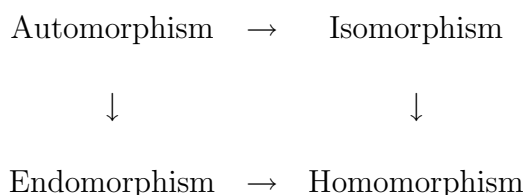


Figure A.2: The mathematical concept of Algebraic Structure

A *cyclic group* is a group isomorphic to either \mathbf{Z} or \mathbf{Z}_n for some n . These groups can be generated by one element. For example \mathbf{Z}_4 is generated by 3:

$$\{0, 3, 3 + 3, 3 + 3 + 3\} = \{0, 3, 6, 9\} \equiv \{0, 3, 2, 1\} \pmod{4} = \mathbf{Z}_4$$

Theorem A.4. *Let G be a finite cyclic group of order n and let $d > 0$ divide n . Then*

- (i) *G has a unique subgroup of order d .*
- (ii) *G has d elements of order dividing d , and G has $\phi(d)$ elements of order exactly d (where $\phi(d)$ is the Euler Totient function).*

Theorem A.5. *A finite abelian group, G , is isomorphic to*

$$\mathbf{Z}_{n_1} \oplus \mathbf{Z}_{n_2} \oplus \dots \oplus \mathbf{Z}_{n_s}$$

with $n_i | n_{i+1}$ for $i = 1, 2, \dots, s - 1$. The n_i are uniquely determined by G .

A.6 Field theory

A field is a set in which we can perform analogues of the operations $(+, -, \times)$ for all elements and also \div by all elements except for 0. We usually think of division by an element as multiplying by that element's inverse. So $b/a = ba^{-1}$ where a^{-1} is the element such that $a^{-1} \times a = 1$. The formal definition of a field follows.

A *field* is a commutative ring $(F, +, \times)$ such that 0 does not equal 1 and all elements of F except 0 have a multiplicative inverse. (Note: 0 and 1 here stand for the identity elements for the $+$ and \times operations, and not the real numbers). This means that the following all hold:

- *Closure of F under $+$ and \times*

For all a, b belonging to F , both $a + b$ and $a \times b$ belong to F (or more formally, $+$ and \times are *binary operations* on F)

- *Both $+$ and \times are associative*

For all $a, b, c \in F$, $a + (b + c) = (a + b) + c$ and $a \times (b \times c) = (a \times b) \times c$.

- *Both $+$ and \times are commutative*

For all a, b belonging to F , $a + b = b + a$ and $a * b = b * a$.

- *The operation \times is distributive over the operation $+$*

For all a, b, c , belonging to F , $a \times (b + c) = (a \times b) + (a \times c)$.

- *Existence of an additive identity*

There exist an element $0 \in F$, such that for all a belonging to F , $a + 0 = a$.

- *Existence of a multiplicative identity*

There exist an element $1 \in F$, different from 0, such that for all a belonging to F , $a * 1 = a$.

- *Existence of additive inverses*

For every $a \in F$, there is an element $-a \in F$, such that $a + (-a) = 0$.

- *Existence of multiplicative inverses*

For every $a \neq 0$ in F , there is an element $a^{-1} \in F$, such that $a \times a^{-1} = 1$.

The requirement $0 \neq 1$ ensures that the set which only contains a single element is not a field.

Let K be a field. There is a ring homomorphism $\varphi : \mathbb{Z} \rightarrow K$ that sends $1 \in \mathbb{Z}$ to $1 \in K$. If φ is injective then we say K has *characteristic 0*. Otherwise there is a smallest positive integer q such that $\varphi(q) = 0$ and we say K has *characteristic q* .

So if we are in a field $(K, +, \times)$ with identities 0 and 1 then consider the elements,

$$1, 1 + 1, 1 + 1 + 1, \dots$$

Now if there is n such that

$$\frac{1 + 1 + \dots + 1}{n \text{ times}} \equiv 0$$

then we say the field K has characteristic n . If however all those elements are unique then we say K has characteristic 0.

(Clearly if K is a finite field then it cannot have characteristic zero, but there are infinite fields with positive characteristic.)

Theorem A.6. *The characteristic q is prime*

Proof. (By Contradiction) Assume $q = ab$ with $1 < a \leq b < q$. Then $\varphi(a)\varphi(b) = \varphi(q) = 0 \Rightarrow \varphi(a) = 0$ or $\varphi(b) = 0$
 \Rightarrow CONTRADICTION so q is prime

- A multiplicative group is formed from a field $K(+, *)$ under the multiplication operator with the zero element removed. This group is usually denoted K^\times .

- When K has characteristic 0 the field \mathbf{Q} of rational numbers is contained in K . When K has characteristic q the field \mathbf{F}_q of integers modulo q is contained in K .
- Let K and L be fields with $K \subseteq L$. If $\alpha \in L$ we say that α is *algebraic over* K if there exists a non-constant polynomial

$$f(X) = X^n + a_{n-1}X^{n-1} + \dots + a_0$$

with $a_0, \dots, a_{n-1} \in K$ such that $f(\alpha) = 0$.

- We say that the field L is *algebraic over* K (or that L is an algebraic extension of K) if every element of L is algebraic over K .
- An *algebraic closure* of a field K is a field \overline{K} containing K such that:
 1. \overline{K} is algebraic over K .
 2. Every non-constant polynomial $g(X)$ with coefficients in \overline{K} has a root in \overline{K} ($\Rightarrow \overline{K}$ is algebraically closed).

If $g(X)$ has degree n and has a root $\alpha \in \overline{K}$, then we can write $g(X) = (X - \alpha)g_1(X)$ with $g_1(X)$ of degree $(n - 1)$. By induction we see that $g(X)$ has exactly n roots (counting multiplicatively) in \overline{K} .

- It can be shown that every field K has an algebraic closure, and that any two algebraic closures of K are isomorphic. Assume that a particular algebraic closure of a field K has been chosen, and refer to it as *the algebraic closure of* K .
- A field K is said to be *algebraically closed* if every polynomial (in one variable of degree at least 1), with coefficient in K , has a zero (root) in K . \mathbf{C} is algebraically closed (by fundamental theorem of algebra).

The *algebraic closure* of K can also be defined as the smallest algebraically closed field containing K .

□

A.6.1 Finite Fields

A *finite field* is a field that contains only finitely many elements. The finite fields are completely known as described below.

1. Every finite field has q^n elements for some prime q and some integer $n \geq 1$.
(This q is the characteristic of the field.)
2. For every prime q and integer $n \geq 1$, there exists a finite field with q^n elements.
3. All fields with q^n elements are isomorphic, which justifies using the same name for all of them, \mathbf{F}_{q^n} (in other literature $\mathbf{GF}(q^n)$ is often used).

Appendix B

Programming code

This appendix contains the related code in Matlab programs for the computation in ECC. It is a function associated with the book by Wade Trappe & Lawrence Washington (2006). The functions are available at

www.prenhall.com/washington

Below is a table summarising on the programs.

Appendix	Code	Description
B.1	addell.m	Addition points on the elliptic curve E
B.2	multell.m	Computing M^{th} multiple of p on the elliptic curve
B.3	multsell.m	Prints the first M multiples of p
B.4	randprime.m	Finds a random prime between 1 and N

We also used Sage to generate graph on ECC but we did not include the code here because Sage is a open-source software that can be downloaded at

www.sagemath.org

B.1 The Matlab code for addell.m

```
function p3 = addell(p1,p2,a,b,n);

% This function add points on the elliptic curve
%  $y^2 = x^3 + ax + b \pmod n$ 
% The points are represented by
% p1(1) = x1    p1(2) = y1
% p2(1) = x2    p2(2) = y2

if (any(p1==Inf)),

    p3=p2;

    return;

end;

if (any(p2==Inf)),

    p3=p1;

    return;

end;

x1=p1(1);

x2=p2(1);

y1=p1(2);

y2=p2(2);

z1=1; % this will store the gcd incase the addition

        produced a factor of n

if ( (x1==x2) & (y1==y2) & (y1==0)), % an infinity case

    p3(1)=inf; p3(2)=inf;

    return;

end;
```

```

if ( (x1==x2) & (y1 ~= y2)),           % an infinity case
    p3(1)=inf; p3(2)=inf;
    return;
end;
if (all(p1==p2) & (gcd(y1,n)~=1) & (gcd(y1,n) ~=n)),
    z1=gcd(y1,n);
    p3=[];
    disp(['Elliptic Curve addition produced a factor of n,
        factor = ',num2str(z1)]);
    return;
end;
if all(p1==p2),
    temp=mod(2*y1,n);
    if temp==0,
        p3(1)=Inf;
        p3(2)=Inf;
        return;
    end;
    den=powermod(2*y1, -1, n);
    num=mod(x1*x1,n);
    num=mod(mod(3*num,n) + a,n);
    m=mod(num*den,n);
    temp=mod(m*m,n);
    x3=mod(temp-x1-x2, n);
    temp=x1-x3;

```

```

y3=mod(m*temp,n);

y3=mod(y3-y1,n);

else % case p1 ~= p2

    if (gcd(x2-x1,n) ~= 1),

        z1=gcd(x2-x1,n);

        p3=[];

        disp(['Elliptic Curve addition produced a factor of n,

            factor= ',num2str(z1)]);

        return;

    end; % end if gcd

temp=mod(x2 - x1,n);

if (mod(n,temp)==0), % Infinity case

    p3(1)=Inf;

    p3(2)=Inf;

    return;

end;

den=powermod(temp,-1,n);

num=mod(y2-y1,n);

m=mod(num*den,n);

temp=mod(m*m,n);

x3=mod(temp-x1-x2, n);

temp=x1-x3;

y3=mod(m*temp,n);

y3=mod(y3-y1,n);

end;

```

$$p_3(1) = x^3;$$

$$p_3(2) = y^3;$$

B.2 The Matlab code for multell.m

```
function y = multell(p,M,a,b,n);

% This function prints the Mth multiple of p on the elliptic
% curve with coefficients a and b mod n.

z1=M;

y=[inf inf];

while (z1 ~=0),

    while (mod(z1,2) ==0),

        z1=(z1/2);

        p=addell(p,p,a,b, n)

        if (length(p)==0),

            y=[];

            disp('Multell found a factor of n and exited');

            z1

            return;

        end;

    end; %end while

    z1=z1-1;

    y=addell(y,p,a,b,n)

    if (length(y)==0),

        disp('Multell found a factor of n and exited');

        z1

        return;

    end;

end;
```

B.3 The Matlab code for multsell.m

```
function y = multsell(p,M,a,b,n);

% This function prints the first M multiples of p

p=p(:)';

y=zeros(M,2);

y(1,:)=p;

q=p;

for k=2:M,

    z=addell(p,q,a,b,n);

    q=z;

    if (length(z)==0), % must have returned a factor!

        y(k:M,:)=[]; % null out the rest

        disp('Multsell ended early since it found a factor');

        return;

    end;

    y(k,:)=z;

end;
```

B.4 The Matlab code for randprime.m

```
function y = randprime(N);

% This function finds a random prime between 1 and N

% The prime is tested using Miller-Rabin

N1=N-1;

flag=1;

while flag,

    y=1+floor((N1)*rand(1,1));

    if primetest(y),

        return;

    end;

end; %end while
```

REFERENCES

- Boneh, D., & Shparlinski, I.E. (2001). Proceedings of Crypto 2001: *On the unpredictability of bits of the Elliptic Curve Diffie Hellman scheme*. Springer-Verlag.
- Diffie, W., & Hellman, M.E. (1976). New Directions in Cryptography. *IEEE Transactions on Information Theory*, 31(4), 469-472.
- Elbirt, A.J. (2009). *Understanding and Applying Cryptography and Data Security*(1st ed.). Boston, MA: Auerbach Publications.
- ElGamal, T. (1985). A Public Key Cryptosystem and a Signature Scheme based in Discrete Logarithm. *IEEE Transactions on Information Theory*, 31(4), 469-472.
- Ertaul, L., & Lu, W. (2005). ECC based Threshold Cryptography for Secure Data Forwarding and Secure Key Exchange in MANET(I). *R. Boutaba et al. (Eds.): NETWORKING 2005, LNCS 3462*, 102-113.
- Forouzan, B.A. (2008). *Cryptography and Network Security* (1st ed.). New York: McGraw-Hill.
- Hankerson, D., Menezes, A., & Vanstone, S. (2003). *Guide to Elliptic Curve Cryptography*. New York: Springer-Verlag.
- Hoffstein, J., Pipher, J., & Silverman, J.H. (1998). NTRU: A Ring-Based Public Key Cryptosystem, *Lecture Notes in Computer Science* (pp. 267-288). New York:

Springer-Verlag.

Johnson, D., & Menezes, A. (1999). The Elliptic Curve Digital Signature Algorithm. Retrieved 19 May 2009, from <http://www.cacr.math.uwaterloo.ca>

Juriscic, A., & Menezes, A.J. (2005). ECC Whitepapers: Elliptic Curves and Cryptography, Certicom corp. Retrived 13 July 2010, from <http://www.certicom.com/research/weccrypt.html>

Kaliski, B.S.Jr. (1997). IEEE P1363: A Standard for RSA, Diffie-Hellman, and Elliptic Curve Cryptography. *Lecture Notes In Computer Science* (pp. 117-118). New York: Springer-Verlag.

Koblitz, N. (1987). Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48, 203-209.

Kurt, Y. (2006). A new key exchange Primitive Based on the Triple Decomposition Problem. *International Association for Cryptologic Research*.

Massey, J.L., & Omura, J.K. (1983). A New Multiplicative Algorithm over Finite Fields and its Applicability in Public Key Cryptography. *EUROCRYPT '83*.

Mao, W. (2003). *Modern Cryptography: Theory and Practise* (8th ed.). Upper Saddle River, NJ: Prentice Hall PTR.

Mel, H.X., & Baker, D.M. (2005). *Cryptography Decrypted* (9th ed.). Boston, MA: Pearson Education, Inc.

Menezes, A.J. (1995). Elliptic Curve Cryptosystems. *RSA Laboratories Crypto-Bytes*, 1(2).

Miller, V. (1986). Uses of Elliptic Curves in Cryptography. *Advances in Cryptology - CRYPTO '85, Lecture Notes in Computer Science* (pp. 417 - 426). New York: Springer-Verlag.

Othman, W.A.M., & Zazali, H.H. (2009). Key Exchange Method using Triple Decomposition Problem in Elliptic Curve. *Proceeding of Simposium Kebangsaan Sains Matematik (SKSM)* (pp. 913 - 916). Melaka: Malaysia.

Pipihier, J., Hoffstein, J., & Silverman, J.H. (1996). NTRU: A New High Speed Public Key Cryptosystem. Manuscript. *Rump Session Crypto '96*.

Rivest, R., Shamir, A., & Adleman, L. (1978). A method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2), 120 - 126.

Rosing, M. (1998). *Implementing Elliptic Curve Cryptography*. Greenwich, CT: Manning Publications Co.

Schneier, B. (1996). *Applied Cryptography: Protocols, Algorithms and Source Code*

in C (2nd ed.). New York: John Wiley & Sons.

Shpilrain, V., & Ushakov, A. (2005). A New Key Exchange Protocol Based on the Decomposition Problem. *International Association for Cryptologic Research*.

Stallings, W. (2006). *Cryptography and Network Security: Principles and Practices* (4th ed.). Upper Saddle River, NJ: Pearson Prentice Hall.

Tork, E.V. (1992). Elliptic curves over Finite Field. *George Mason University*.

Trappe, W., & Washington, L.C. (2006). Introduction to Cryptography with Coding Theory (2nd ed.). Upper Saddle River, NJ: Pearson Prentice Hall.

Vanstone, S. (1992). Responses to NIST's Proposal. *Communication of the ACM*. 35, 50-52.

Wagner, N.R. (2003). *The Laws of Cryptography with Java Code*. Retrieved 11 November 2009, from <http://www.cs.utsa.edu/wagner/laws/Rabin.html>

Washington, L.C. (2003). *Elliptic Curves: Number Theory and Cryptography*. Boca Raton, FL: Chapman & Hall/CRC. Inc.

Zazali, H.H., & Othman, W.A.M. (2009). New Key Exchange in Elliptic Curve based on the Decomposition Problem. *Proceeding of The 5th International Conference on Mathematics, Statistic and their applications*. Padang: Indonesia.