

THE LIBYAN ACADEMY



**School of Applied Sciences and Engineering
Department of Electrical and Computer Engineering**

Cryptography Based on Elliptic Curve

Submitted by

Mahmoud Hassan Said

Thesis submitted to the department of Electrical and Computer Engineering in conformity

with the department requirements as partial fulfillment for the degree of

M.Sc. in Engineering

Supervisor by

Dr. Ibrahim A. Almerhag

January-2015

DECLARATION

I declare that this thesis which I now submit for the examination for the award of M.Sc. in Engineering is entirely my own work and has not been submitted in whole or part for an award in any other university or institute.

This thesis was prepared according to the regulations for postgraduate study of the Libyan Academy.

DEDICATION

To my mother and in memory of my father and my grandparents with thanks and gratitude, to my family on whole and my teachers since I started to study for their encouragements, supports and advices.

ACKNOWLEDGEMENTS

Praise be to Allah almighty. Writing an acknowledgement is not an easy. This is because there are so many people who have helped along the way. This thesis could not have reached this final outcome without the crucial help of my supervisor **Dr. Ibrahim A. Almerhag** , so that, I would like to express my sincere gratitude to him for his constant support of my varied research interests and relentless optimism.

I also extend my sincere appreciation to the members of my family.

Mahmoud Hassan Said

Engmhsz@Gmail.com

January 2015, Tripoli, Libya

Table of Contents

Subject	Page no.
DECLARATION	II
DEDICATION	III
ACKNOWLEDGEMENT	IV
TABLE OF CONTENTS	VI
LIST OF TABLES	VIII
LIST OF FIGURES	IX
LIST OF SYMBOLS AND ABBREVIATIONS	X
ABSTRACT (IN ARABIC)	XII
ABSTRACT (IN ENGLISH)	XIII
Chapter One: INTRODUCTION	
1. Information Hiding.....	2
1.1 Cryptography	3
1.1.1 Symmetric Cryptosystems	4
1.1.2 Asymmetric Cryptosystems	4
1.2 Problem Statement	5
1.3 Methodology.....	6
1.4 Theses Outline.....	6
Chapter Two: Understanding of Public-Key Cryptosystems	
2. Public Key Cryptography	9
2.1 Discrete Logarithm Problem	11
2.2 Elliptic Curve Discrete Logarithm Problem	12
2.3 The RSA Cryptosystem	12
2.4 Public-Key Cryptosystems Based on the Discrete Logarithm Problem .	13
2.4.1 Diffie–Hellman Key Exchange.....	14
2.4.2 ElGamal's Public-Key Cryptosystems.....	15
2.5 Elliptic Curve Cryptosystems.....	16
2.5.1 Elliptic Curves over F_p	16
2.5.2 Arithmetic Operations on Elliptic Curves	17
2.5.3 Elliptic Curved Discrete Logarithm Problem.....	20
2.5.4 Elliptic Curve: Some Definitions.....	21
2.5.5 Elliptic Curve Cryptography (ECC) domain parameters.....	21
2.5.6 Diffie–Hellman Key Exchange with Elliptic Curves.....	22
2.5.7 Security of Elliptic Curve Cryptography.....	23
Chapter Three: Proposed Solution	
3. Elliptic Curve Cryptosystems.....	25
3.1 Massey-Omura Elliptic Curve Cryptosystem.....	25
3.2 ElGamal Elliptic Curve Cryptography.....	27
3.3 Menezes–Vanstone Elliptic Curve Cryptography.....	28
3.4 Al-Saffar and Said's Algorithm	30
3.5 Proposed Methods of MVECC.....	32
Chapter Four: Results And Discussion	
4.1 Experimental Setup.....	38
4.2 Obtained Results.....	39

4.3 Analysis and Discussions.....	40
4.3.1 Computational Complexity.....	40
4.3.2 Execution Time for Encryption.....	44
4.3.3 Execution Time for Decryption.....	44
4.3.4 Execution Time for Encryption and Decryption.....	45
Chapter Five: Conclusions	
5.1 Conclusions.....	48
5.2 Suggestions for future work	49
<i>REFERENCES</i>	

List Tables

Table	Page
Table (2.1): Applications for Public-Key Cryptosystems.....	11
Table (2.2): Comparable Key Sizes in Terms of Computational Effort for Cryptanalysis.....	23
Table (3.1): The points on $y^2 = x^3 + 2x + 2 \pmod{17}$	26
Table (4.1): Execution Time (Milliseconds) of Encryption of different text file size.....	39
Table (4.2): Execution Time (Milliseconds) of Decryption of different text file size.....	39
Table (4.3): Execution Time (Milliseconds) of Encryption and Decryption of different text file size.....	40

List Figures

Figure	Page
Figure 1.1: Symmetric key cryptosystem	4
Figure 1.2: Public-key (asymmetric) cryptosystem	5
Figure 2.1: $y^2 = x^3 - 3x + 3$ over real numbers.....	17
Figure 2.2: Point addition on an elliptic curve over the real numbers.....	18
Figure 2.3: Point doubling on an elliptic curve over the real numbers.....	18
Figure 2.4: The inverse of a point P on an elliptic curve	20
Figure 4.1: Execution time for encryption (in milliseconds).....	44
Figure 4.2: Execution time for decryption. (in milliseconds).....	44
Figure 4.3: Execution time for encryption and decryption (in milliseconds).....	45

List of Symbols and Abbreviations

<i>DSA</i>	<i>Digital Signature Algorithm</i>
<i>RSA</i>	<i>Rivest-Shamir-Adleman</i>
<i>ECDLP</i>	<i>Elliptical Curve Discrete Log Problem</i>
<i>ECC</i>	<i>Elliptic Curve Cryptography</i>
<i>DEA</i>	<i>Data Encryption Algorithm</i>
<i>DES</i>	<i>Data Encryption Standard</i>
<i>Triple-DES</i>	<i>Triple-Data Encryption Standard</i>
<i>IDE</i>	<i>Integrated Development Environment</i>
<i>MVECC</i>	<i>Menezes–Vanstone Elliptic Curve Cryptography</i>
$\mathbb{Z}_p^*$	<i>Referred to as a prime field</i>
<i>DSA</i>	<i>Digital Signature Algorithm</i>
<i>ECDH</i>	<i>Elliptic Curve Diffie–Hellman key exchange</i>
<i>ECDSA</i>	<i>Elliptic Curve Digital Signature Algorithm</i>
<i>DLP</i>	<i>Discrete logarithm problem</i>
$\mathbb{Z}_n$	<i>ring for any positive integer n</i>
$\mathbb{F}_p$	<i>Finite Field</i>
$\Phi(n)$	<i>Euler phi function of n, the number of integers in $\mathbb{Z}_n$ that are relatively prime to n.</i>
<i>NIST</i>	<i>National Institute of Standards and Technology</i>
<i>FIPS</i>	<i>Federal Information Processing Standards</i>
$\#E$	<i>The order of curve E</i>

<i>O</i>	<i>Point at infinity (on an elliptic curve)</i>
<i>E</i>	<i>Elliptic Curve</i>

الخلاصة

اصبح الاهتمام بعلم التشفير شيء مهم في حياتنا حتى نستطيع حفظ المعلومات بشكل امن. ويعرف التشفير بأنه هو عملية تحويل النص العادي إلى نص مشفر وفك التشفير هو عملية الحصول على لرسالة الأصلية من النص المشفر.

يمكن استخدام المنحنى الاهليجي (ECC) لتطوير مجموعة متنوعة من برامج التشفير ويمكننا بالعديد من العمليات منها تبادل المفاتيح، والتشفير والتوقيع الرقمي.

ويمتاز التشفير باستخدام المنحنى الاهليجي لوقورن بنظام RSA انه يستخدم عدد اقل من البت وبنفس مستوى الامن لذا يمتاز بالسرعة واستهلاك اقل للطاقة والذاكرة ، حيث اصبح يستخدم على نطاق واسع في تطبيقات الهواتف المحمولة.

في بحثنا هذا تم دراسة النظام المسمى مينيزيس-فانستون للتشفير (MVECC). حيث تم بإضافة بعض التعديلات لهذا النظام لغرض تحسين الأداء والسرعة ، وتم تطوير نسختين مختلفتين من هذا النظام. وقد تم كتابة برنامج المحاكاة لتقييم اداء الخوارزميات المقترحة ومقارنتها من النظام (MVECC) من ناحية الكفاءة وسرعة انجاز عمليتي التشفير وفك التشفير للنظام بالإضافة الى حساب التعقيد الزمني لكل منهم.

Abstract

The need for information security leads to the evolution of cryptography. In other words, cryptography is the science of keeping information secure. It involves encryption and decryption of messages. Encryption is the process of converting a plain text into cipher text and decryption is the process of getting back the original message from the encrypted text.

Elliptic curve arithmetic can be used to develop a variety of elliptic curve cryptographic (ECC) schemes including key exchange, encryption and digital signature. There have been various cryptographic algorithms suggested.

The principal attraction of elliptic curve cryptography compared to RSA is that it offers equivalent level of security for a smaller key-size. The resulting elliptic curve cryptographic systems provide relatively small block size, high speed, and high security. ECC helps to establish equivalent security with lower computing power and battery resource usage, it is becoming widely used for mobile applications.

In this study the Menezes–Vanstone elliptic curve cryptography (MVECC) has been studied and analyzed. In addition, attempts to improve it by suggesting a new two variations on MVECC to make encryption more efficient than the original elliptic curves cryptosystems. Simulation software was written to assess the behavior all of these proposed algorithms, and we compared them with the original schemes MVECC according to the complexity and the time which they took to implement each scheme.

Chapter One

Introduction

1. Information Hiding

Today most of communication occurs electronically there have been advancements utilizing digital multimedia signals as vehicles for stenographic communication these signals, which are typically audio, video or still imagery, are cover signals. Schemes where the original cover signal is needed to reveal the hidden information are known as cover escrow[1]. Several techniques to hide information are exist: steganography, watermarking, and cryptography, are well known and widely used to hide the original message.

In steganography the information is hidden in the image whereas watermarking actually adds something to the image (like the word "Confidential"), and therefore becomes part of the image. Some people might consider steganography to be related to encryption, but they aren't the same thing. We use encryption the technology to translate something readable to something unreadable to protect sensitive or confidential data. In steganography the information is not necessarily encrypted, just hidden from plain view, while cryptography is the art of protecting information by transforming (encrypting) it into an unreadable format, called cipher text.

Steganography is a technique for hiding a secret message within a larger one in such a way that others cannot discern the presence or contents of the hidden message[2]. The term steganography refers to the art of covert communications. The message is embedded within another object known as a cover work, by tweaking its properties. The resulting output is known as steganography.

Watermarking: digital watermarking is a technique for inserting information (the watermark) into an image (visible or invisible). The idea is to change the text into a format which is not easy to decrypt without decryption key changing the alphabets with another alphabets or make a key to arrange the alphabets.

Cryptography enables you to store sensitive information or transmit them across insecure network (like the Internet) so that it cannot be read by anyone except the intended recipient. Cryptography is an important tool used for thousands of years for securing transactions. It was originally intended to disguise messages so that adversaries could not acquire or alter sensitive information. Cryptography involves encryption, which is the conversion of plaintext to an unreadable format known as ciphertext. If the cryptosystem is secure; only an entity with the secret key can decrypt the ciphertext and convert it back to

plaintext. Since its origination, cryptography has been divided to include two types of cryptosystems: symmetric and asymmetric. These terms define whether the cryptosystem uses a single key or a pair of keys for encryption and decryption [2].

1.1 Cryptography

There have been many known cryptographic algorithms. The heart of any cryptographic algorithm is the “key” used for encrypting/decrypting the information. Many of the cryptographic algorithms are available publicly. The general method is in using a publicly known algorithm while maintaining the key a secret [3].

Based on the key cryptosystems can be classified into two categories: Symmetric and Asymmetric.

- In symmetric key cryptosystems, we use the same key for both encryption as well as the corresponding decryption.
- Asymmetric or public key or shared key cryptosystems use two different keys. One is used for encryption while the other key is used for decryption. The two keys can be used interchangeably. One of the keys is made public (shared) while the other key is kept a secret.

In general, symmetric key cryptosystems suffer from key management. However, they are preferred over public key systems due to the following reasons:

1. Ease of computation.
2. Smaller key length providing the same amount of security as compared to a larger key in Public key systems.

The idea of using Elliptic Curves in Cryptography was introduced by Victor Miller and Neal Koblitz as an alternative to established public-key systems such as Digital Signature Algorithm (DSA) and Rivest-Shamir-Adleman (RSA). The Elliptical Curve Discrete Log Problem (ECDLP) makes it difficult to break an Elliptic Curve Cryptography (ECC) as compared to RSA and DSA where the problems of factorization or the discrete log problem can be solved in sub-exponential time. This means that significantly smaller parameters can be used in ECC than in other competitive systems such as RSA and DSA. This helps in having smaller key size hence faster computations.

1.1.1 Symmetric Cryptosystems

Symmetric algorithms share the same key for encryption and decryption. Figure 1.1 shows the symmetric key cryptosystem. Keeping this single key private is essential to the security of the algorithms. These traditional cryptosystems suffer from many weaknesses including key exchange and key management (when large groups must manage many pairs of keys). Despite this security risk, they do offer the advantages of having a small key length and consuming a low amount of computing power. Examples of well-known symmetric algorithms include the Data Encryption Algorithm (DEA) defined by the Data Encryption Standard (DES), and Triple-DES.

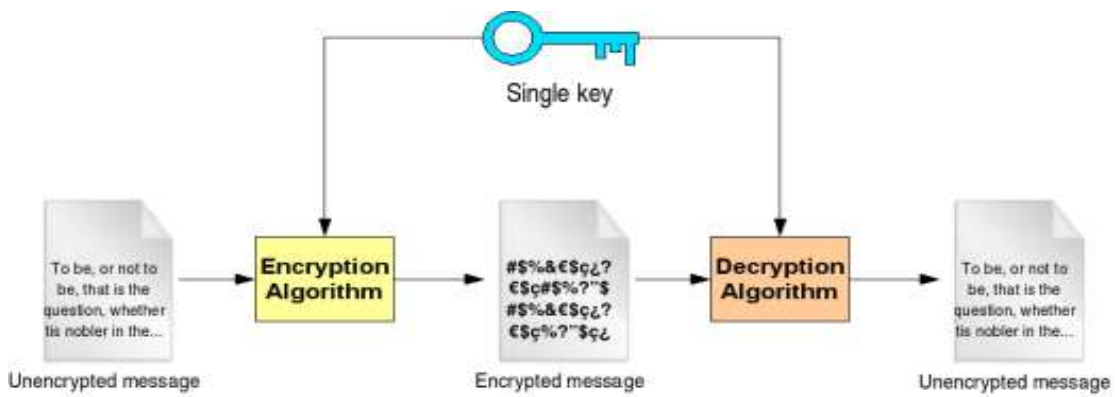


Figure 1.1: Symmetric key cryptosystem

1.1.2 Asymmetric Cryptosystems

Asymmetric cryptosystems are better suited for real-world environments where the secure transfer of the secret key is not essential to the security of the system. These algorithms use a pair of keys: one private key typically used for decryption and one public key typically used for encryption. Figure 1.2 shows the asymmetric key cryptosystem. This allows for anyone who knows an entity's public key to send them an encrypted message with assurance that only that entity can decrypt it with their secret private key. The security and utility of public key cryptosystems is aided by their asymmetric design. Well-known asymmetric algorithms include DSA, RSA, and ELGAMAL.

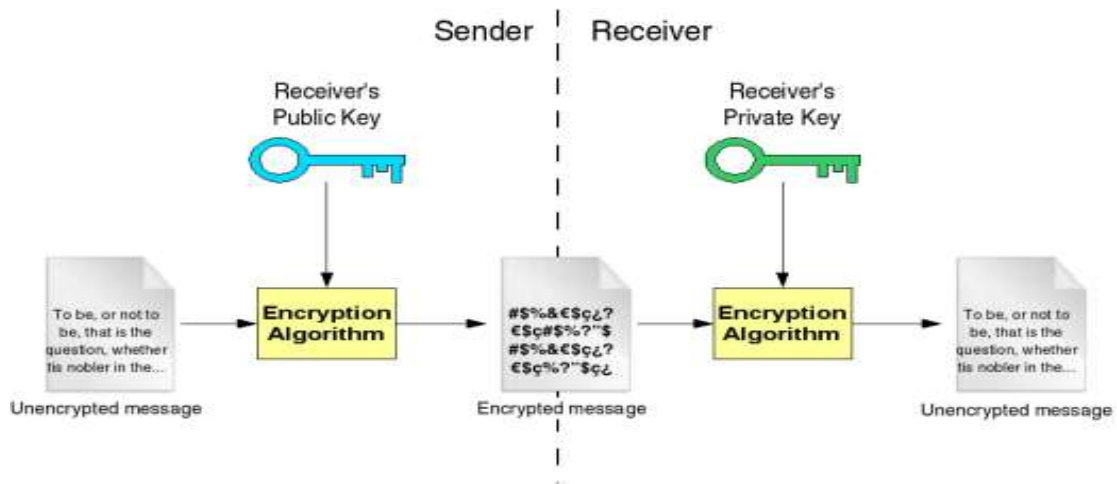


Figure 1.2: Public-key (asymmetric) cryptosystem

1.2 Problem Statement

Asymmetric schemes like RSA and ElGamal require exponentiations in integer rings and fields with parameters of more than 1000 bits, which makes high computational effort on microprocessors with 32-bit or 64-bit arithmetic.

In recent years, Elliptic Curve Cryptography (ECC) has attracted the attention of researchers and product developers due to its robust mathematical structure and highest security. In 2013 a research by Al-Saffar and Said [3] firstly discussed ElGamal and MVECC which were considered as the original schemes in ECC. Secondly it discussed all propositions which have been introduced in Master's thesis written by Al-Saffar, in 2005 [4]. Finally, it compared them with the original schemes according to the complexity and the time which they took to implement each scheme. The conclusion of this research was that ElGamal cryptosystem is dependent on the additive operation on elliptic curve group. If the sender wants to send any message to the receiver, the sender must use the public key of receiver (as the other public key cryptosystem), and in a way, it can change it with the same complexity, because the addition and subtraction have the same computational complexity. However, the MVECC is a very important public key cryptosystem because of the following:

- It does not depend on additive operation on elliptic curve group.
- The message need not be a point on elliptic curve.

In 2009 [5], Ahmed Sadiq and Nasreen Kadhim presented research that introduces a brief overview of the MVECC, and attempts to improve it by suggesting a new variation on MVECC to make encryption more efficient than the original elliptic

curves cryptosystems. The conclusion of this research was the Menezes-Vanstone scheme has advantage that it does not need encoding the plaintext message in the elliptic curve. This makes it more efficient than the original ElGamal scheme. However, it needs computing the inverse of the two numbers in the key point. The plain text consist of two numbers, the cipher text and the key, i.e., (k_1, k_2) . This characteristic exploited to make the system more efficient. Also it proposed new variations of the MVECC. This variation gives the system more security because it mixes between the two numbers of the two-plaintext unit, and it needs calculating the inverses operation once only.

In this research, two versions of MVECC algorithms were proposed. The suggested modification depend on the difficulty of ECDLP to guarantee the same level of security.

1.3 Methodology

In this study, comparison will be used to evaluate the solutions. These comparisons are to compare the execution times of all algorithms used. Where a simulation program will be written using Java platform and using Netbeans integrated development environment (IDE). As the latest version of this Java Platform includes classes and interfaces that makes ECC available to programmers. Namely the Bouncy Castle Crypto package which is a Java implementation of cryptographic algorithms will be used.

After writing the simulation software for each of these algorithms many text files having different sizes will be used to calculate the elapsed time of encryption and decryption process that the algorithm takes, where the algorithm which takes shorter time is the better algorithm in terms of execution speed. Then the complexity of the encryption and decryption functions for the MVECC and the two proposed methods will be computed and compared.

1.4 Theses Outline

The rest of this thesis is divided into five chapters, which will be discussed here one by one. Chapter two introduces the theme in a general way; in order to allow a diversified public to understand, what is the public key cryptography and its description. In addition to different types of asymmetric cryptosystem algorithms.

Chapter three gives more details and explains implemented mechanisms to cryptography based on elliptic curve. Moreover this chapter explains some algorithms that rely on the idea of a elliptic curve and some related problems. In addition, this chapter presents an explanation of Menezes-Vanstone Elliptic Curve Cryptography algorithm and the proposed solution to improve the performance of this algorithm.

Chapter four details the implementation of the proposed solutions. Where this chapter talks about java packages and classes used in this implementation, and the parameters that will be used for comparison to get the results. In addition, this chapter, discusses the results, where the time complexity of the algorithm MVECC will be calculated, the previous proposal in [3] and the proposed algorithms then compute the time elapsed in the encryption and decryption process and finally compare the results.

In the last chapter the research conclusions and suggestions for further improvement are given.

Chapter Two

Understanding of Public-Key Cryptosystems

2. Public Key Cryptography

In a public key or asymmetric cryptosystem there are two keys. The public key is published in a directory and allows encryption. And the private key is kept secret and allows decryption. The proposal of public key cryptosystems in 1976 by Whitfield Diffie and Martin Hellman introduced a revolutionary way to address modern security issues such as key management, authentication, non-repudiation, and signatures in a digital environment.

All cryptosystems are secure only if the difficulty of the mathematical problem that they are based on is determined to be hard. There are only three major families of public-key algorithms which are of practical relevance. They can be classified based on their underlying computational problem as [2]:

1. Integer-Factorization Schemes; several public-key schemes are based on the fact that it is difficult to factor large integers. The most prominent representative of this algorithm family is RSA.
2. Discrete Logarithm Schemes; there are several algorithms which are based on what is known as the discrete logarithm problem in finite fields. The most prominent examples include the Diffie–Hellman key exchange, Elgamal encryption or the Digital Signature Algorithm (DSA).
3. Elliptic Curve Schemes; a generalization of the discrete logarithm algorithm are elliptic curve public-key schemes. The most popular examples include Elliptic Curve Diffie–Hellman key exchange (ECDH) and the Elliptic Curve Digital Signature Algorithm (ECDSA).

Discrete logarithm problem (DLP) and the elliptic curve discrete logarithm problem (ECDLP) are discussed in the following sections.

Asymmetric algorithms rely on one key for encryption and a different but related key for decryption. These algorithms have the important characteristic it is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key.

A public-key encryption scheme has ingredients as the following:

1. Plaintext: is the readable message or data that is fed into the algorithm as input.

2. Encryption algorithm: the encryption algorithm performs various transformations on the plaintext.
3. Public and private keys: is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public or private key that is provided as input.
4. Ciphertext: is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.
5. Decryption algorithm: accepts the ciphertext and the matching key and produces the original plaintext.

The essential steps are the following:

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private. Each user maintains a collection of public keys obtained from others.
3. If Bashir wishes to send a confidential message to Amina, Bashir encrypts the message using Amina's public key.
4. When Amina receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Amina knows her private key.

With this approach, all participants have access to public keys, and private keys are generated locally by each participant and therefore need never be distributed. As long as a user's private key remains protected and secret, incoming communication is secure. At any time, a system can change its private key and publish the companion public key to replace its old public key.

Public-key systems are characterized by the use of a cryptographic algorithm with two keys, one held private and one available publicly. Depending on the application, the sender uses either the sender's private key or the receiver's public key, or both, to perform some type of cryptographic function. In broad terms, we can classify the use of public-key cryptosystems into three categories [2]:

1. Encryption/decryption: The sender encrypts a message with the recipient's public key.

2. Digital signature: The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.
3. Key exchange: Two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parties.

Some algorithms are suitable for all three applications, whereas others can be used only for one or two of these applications. Table 2.1 indicates the applications supported by some famous algorithms.

Table (2.1) Applications for Public-Key Cryptosystems

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSA	No	Yes	No

2.1 Discrete Logarithm Problem

The discrete logarithm problem, like the factoring problem, is said to be difficult in addition to being the hard direction of a one-way function [2]. It involves the mathematical structure of groups, of which the simplest definition is a nonempty set G together with a binary operation. Let g be an element of group G . If the order n of element g is also the order of the group G , then g is referred to as a generator of G . This means that repeated exponentiation of g ($g * g * g$) will yield all elements of G . Cryptography commonly uses the group Z_p^* , a finite and multiplicative group of integers modulo a prime number p . The discrete logarithm problem is characterized by these elements under the following condition:

Given a generator g of the multiplicative group Z_p^* , a prime integer p , and another element $h \in Z_p^*$, find the unique integer j in the interval $[1, p - 1]$ such that $h \equiv g^j \pmod{p}$ [6].

The inverse operation of exponentiation on $g^j \pmod{p}$ can be computed efficiently, but it is not practical to calculate $g^j \pmod{p}$ for large values. This is why the DLP is a one-way function and considered a hard problem.

2.2 Elliptic Curve Discrete Logarithm Problem

Based on the infeasibility of computing discrete logs on elliptic curves over finite fields, the elliptic curve discrete logarithm problem is the security behind elliptic curve cryptography. Cryptosystems based on the computational complexity of this mathematical problem include ECDSA, ECElGamal, and elliptic curve Diffie-Hellman. ECDLP is similar to the aforementioned discrete logarithm problem and can be thought of as an analogue to DLP. In the ECDLP, however, the subgroup $\mathbf{Z}_p^*$ is replaced by the group of points on an elliptic curve over a finite field. ECDLP is considered to be significantly harder than the DLP, thus giving elliptic curve cryptosystems a greater strength-per-key-bit than their non-analogue discrete logarithm counterparts.

2.3 The RSA Cryptosystem

The RSA crypto scheme, sometimes referred to as the Rivest–Shamir–Adleman algorithm, is currently the most widely used asymmetric cryptographic scheme. In 1977, Ronald Rivest, Adi Shamir and Leonard Adleman proposed a scheme.

The RSA scheme is a block cipher in which the plaintext and ciphertext are integers between $\mathbf{0}$ and $\mathbf{n} - \mathbf{1}$ for some $\mathbf{n}$. RSA encryption and decryption is done in the integer ring $\mathbf{Z}_n$ and modular computations play a central role. RSA encrypts plaintexts $\mathbf{x}$, where we consider the bit string representing $\mathbf{x}$ to be an element in $\mathbf{Z}_n = \{0, 1, \dots, \mathbf{n} - 1\}$.

The scheme use of an expression with exponentials plaintext is encrypted in blocks, with each block having a binary value less than some number $\mathbf{n}$. That is, the block size must be less than or equal to $\log_2(\mathbf{n})$.

Encryption and decryption are of the following steps, for some plaintext block $\mathbf{M}$ and ciphertext block $\mathbf{C}$:

$$\mathbf{C} = \mathbf{M}^e \bmod n$$

$$\mathbf{M} = \mathbf{C}^d \bmod n = (\mathbf{M}^e)^d \bmod n = \mathbf{M}^{ed} \bmod n$$

The preceding relationship holds if $\mathbf{e}$ and $\mathbf{d}$ are multiplicative inverses modulo $\Phi(\mathbf{n})$, where $\Phi(\mathbf{n})$ is the Euler's Phi function defined as the number of positive integers less than $\mathbf{n}$ and relatively prime to $\mathbf{n}$ [7].

Both sender and receiver must know the value of $\mathbf{n}$. The sender knows the value of $\mathbf{e}$, and only the receiver knows the value of $\mathbf{d}$. Thus, this is a public-key encryption

algorithm with a public key $\{e, n\}$ and a private key $\{d, n\}$. For this algorithm to be satisfactory for public-key encryption, the following requirements must be met:

1. It is possible to find values of e, d, n such that $M^{ed} \bmod n = M$ for all $M < n$.
2. It is relatively easy to calculate $M^e \bmod n$ and C^d for all values of $M < n$.
3. It is infeasible to determine d given e and n .

RSA Key Generation consists of the following steps:

1. Choose two large primes p and q .
2. Compute $n = p \cdot q$.
3. Compute $\Phi(n) = (p - 1)(q - 1)$.
4. Select the public exponent e is relatively prime to $\Phi(n)$ and $1 < e < \Phi(n)$.
5. Compute the private key d such that $d \cdot e \equiv \text{mod } \Phi(n)$.

The private key consists of $\{d, n\}$ and the public key consists of $\{e, n\}$. Suppose that Amina has published its public key and that Bashir wishes to send the message M to Amina. Then Bashir calculates $C = M^e \bmod n$ and transmits C . On receipt of this ciphertext, Amina decrypts by calculating $M = C^d \bmod n$.

2.4 Public-Key Cryptosystems Based on the Discrete Logarithm Problem

Discrete logarithms are fundamental to a number of public-key algorithms, including Diffie-Hellman key exchange and the digital signature algorithm.

Definition Discrete Logarithm Problem (DLP) in Z_p^* where p is a prime given is the finite cyclic group Z_p^* of order $p - 1$ and a primitive element $\alpha \in Z_p^*$ and another element $\beta \in Z_p^*$. The DLP is the problem of determining the integer $1 \leq x \leq p - 1$ such that $\alpha^x \equiv \beta \bmod p$ [6].

An integer x must exist since α is a primitive element and each group element can be expressed as a power of any primitive element. This integer x is called the discrete logarithm of β to the base α and we can formally write:

$$x = \log_{\alpha} \beta \bmod p \dots\dots\dots(2.1)$$

Computing discrete logarithms modulo a prime as in equation (2.1) is a very hard problem if the parameters are sufficiently large. Since exponentiation $\alpha^x \equiv \beta \bmod p$ is computationally easy, this forms a one-way function that is, a function that is easy to compute in one direction and hard to compute in other direction.

2.4.1 Diffie–Hellman Key Exchange

The Diffie–Hellman key exchange (DHKE), proposed by Whitfield Diffie and Martin Hellman in 1976 [2]. DH is a “key exchange” algorithm because it can only be used to establish a shared secret, which is generally then used as a shared symmetric key. It’s worth emphasizing that the words “Diffie–Hellman” and “key exchange” always go together DH is not for encrypting or signing, but instead it allows users to establish a shared secret. This is no mean feat, since the key establishment problem is one of the fundamental pitfalls to symmetric key cryptography [6].

The security of DH relies on the computational difficulty of the DLP. The basic idea behind the DHKE is that exponentiation in $\mathbf{Z}_p^*$, p prime, is a one-way function and that exponentiation is commutative, i.e., $\mathbf{k} = (\alpha^x)^y \equiv (\alpha^y)^x \bmod p$

The value $\mathbf{k} = (\alpha^x)^y \equiv (\alpha^y)^x \bmod p$ is the joint secret which can be used as the session key between the two parties.

Let us now consider how the Diffie–Hellman key exchange protocol over $\mathbf{Z}_p^*$ works. In this protocol we have two parties, Amina and Bashir, who would like to establish a shared secret key. There is possibly a trusted third party that properly chooses the public parameters which are needed for the key exchange. However, it is also possible that Amina or Bashir generate the public parameters. Strictly speaking, the DHKE consists of two protocols, the set-up protocol and the main protocol, which performs the actual key exchange.

The set-up protocol consists of the following steps:

1. Choose a large prime p .
2. Choose an integer $\alpha \in \{2, 3, \dots, p - 1\}$.
3. Publish p and α .

These two values are sometimes referred to as domain parameters. If Amina and Bashir both know the public parameters p and α computed in the set-up phase, they can generate a joint secret key $\mathbf{k}$ with the following key-exchange protocol:

Amina choose $\mathbf{a} \in \{2, \dots, p - 2\}$.

Amina compute $\mathbf{A} = \alpha^{\mathbf{a}} \bmod p$ and send it to Bashir.

Bashir choose $\mathbf{b} \in \{2, \dots, p - 2\}$.

Bashir compute $\mathbf{B} = \alpha^{\mathbf{b}} \bmod p$ and send it to Amina .

Amina computes $B^a \bmod p$.

Bashir computes $A^b \bmod p$.

Amina and Bashir in fact compute the same session key.

Amina computes $B^a \equiv (\alpha^b)^a \equiv \alpha^{ab} \bmod p$.

While Bashir computes $A^b \equiv (\alpha^a)^b \equiv \alpha^{ab} \bmod p$.

and thus Amina and Bashir both share the session key $KAB \equiv \alpha^{ab} \bmod p$. The key can now be used to establish a secure communication between Amina and Bashir, e.g., by using KAB as key for a symmetric algorithm like AES or Triple-DES.

2.4.2 ElGamal's Public-Key Cryptosystems

The Elgamal encryption scheme was proposed by Taher Elgamal in 1985 [2]. It is also often referred to as Elgamal encryption. its security is also based on the intractability of the discrete logarithm problem and the Diffie.Hellman problem. We consider the Elgamal encryption scheme over the group $\mathbb{Z}_p^*$, where p is a prime. We consider two parties, Amina and Bashir. If Amina wants to send an encrypted message M to Bashir, both parties first perform a Diffie.Hellman key exchange to derive a shared key k . For this we assume that a large prime p and a primitive element α have been generated. Now, the new idea is that Amina uses this key as a multiplicative mask to encrypt M , $C \equiv M * k \bmod p$.

The following is performed assuming that Amina wants to send a message M to Bashir, and $M \in \mathbb{Z}_p^*$ (equivalent to the actual plaintext).

ElGamal Key Generation consists of the following steps:

1. Bashir chooses a large random prime p and a primitive root α modulo p .
2. Bashir then chooses a random integer b with $2 \leq b < p - 1$ and computes $\alpha^b \bmod p$.
3. Bashir's public key is (p, α, α^b) and his private session key is b .

Enciphering stage consists of the following steps:

1. Amina obtains Bashir's public key (p, α, α^b) .
2. She chooses a random natural number a with $2 \leq a < p - 1$.

3. She computes $\alpha^a \bmod p$ and $m\alpha^{ba} \bmod p$.
4. Amina then sends the cipher text $c = (\alpha^a, m\alpha^{ba})$ to Bashir.

Deciphering stage consists of the following steps:

1. Bashir uses his private key to compute $(\alpha^a)^{-b} \bmod p$.
2. Then he decipheres m by computing $(\alpha^a)^{-b} \cdot m\alpha^{ba} \bmod p$.

2.5 Elliptic Curve Cryptosystems

Elliptic curves have been studied for many years and there is an enormous amount of literature on the subject. In 1985, Neal Koblitz and V. S. Miller independently proposed using them for public-key cryptosystems. They did not invent a new cryptographic algorithm with elliptic curves over finite fields, but they implemented existing public-key algorithms, like Diffie-Hellman, using elliptic curves [7]. Elliptic curves are interesting because they provide a way of constructing “elements” and “rules of combining” that produce groups. These groups have enough familiar properties to build cryptographic algorithms.

2.5.1 Elliptic Curves over F_p

The finite field $F(p)$, referred to as a prime field, consists of integers in the interval $[0, \dots, p - 1]$ where p is a prime number. The arithmetic of this field is modulo p so that any calculation results will fall into the finite space defined.

An elliptic curve $E(F_p)$ over a finite field F_p is defined by the following equation:

$$y^2 \equiv x^3 + a \cdot x + b \bmod p \dots\dots\dots(2.2)$$

Where $a, b \in F_p$ and the condition $4 \cdot a^3 + 27 \cdot b^2 \neq 0 \bmod p$, together with an imaginary point of infinity O , create a group on the curve.

The elliptic curve includes all points (x, y) which satisfy the elliptic curve equation modulo p (where x and y are numbers in F_p).

The definition of elliptic curve requires that the curve is nonsingular. Geometrically speaking, this means that the plot has no self-intersections or vertices, which is achieved if the discriminant of the curve $-16(4a^3 + 27b^2)$ is nonzero.

For cryptographic use we are interested in studying the curve over a prime field as in the definition. However, if we plot such an elliptic curve over F_p , we do not get anything remotely resembling a curve. However, nothing prevents us from taking an elliptic curve equation and plotting it over the set of real numbers.

In Figure 2.1 the elliptic curve $y^2 = x^3 - 3x + 3$ is shown over the real numbers.

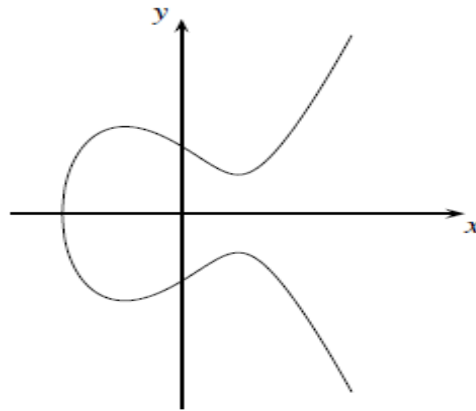


Figure 2.1 $y^2 = x^3 - 3x + 3$ over real numbers

2.5.2 Arithmetic Operations on Elliptic Curves

Addition of points within an elliptic curve group will give another point on the curve, and all multiples of points within the group are also contained on the elliptic curve. There are three rules that addition of points within an elliptic curve group adhere to:

1. $O + O = O$
2. $(x, y) + O = (x, y)$
3. $(x, y) + (x, -y) = O$

- **Addition of Distinct Points**

This is the case where we compute $R = P + Q$ and $P \neq Q$. The construction works as follows: Draw a line through P and Q and obtain a third point of intersection between the elliptic curve and the line. Mirror this third intersection point along the x-axis.

This mirrored point is, by definition, the point R . Figure 2.2 shows the point addition on an elliptic curve over the real numbers.

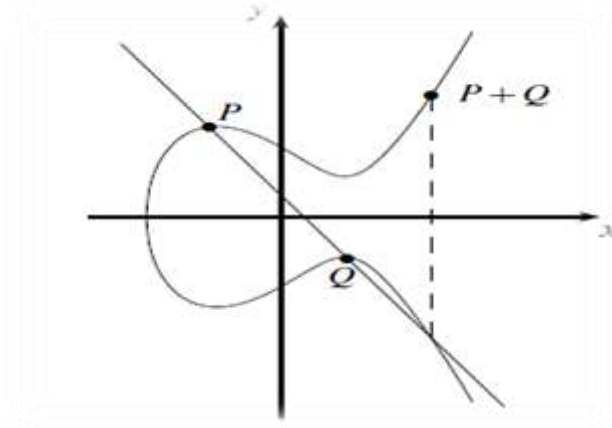


Figure 2.2: Point addition on an elliptic curve over the real numbers

P and Q are distinct if P and $-Q$ are not the same ($x_1 \neq x_2$).

Doubling of a Point

This is the case where we compute $P + Q$ but $P = Q$. Hence, we can write $R = P + P = 2P$. We need a slightly different construction here. We draw the tangent line through P and obtain a second point of intersection between this line and the elliptic curve. We mirror the point of the second intersection along the x-axis. This mirrored point is the result R of the doubling. Figure 2.3 shows the doubling of a point on an elliptic curve over the real numbers.

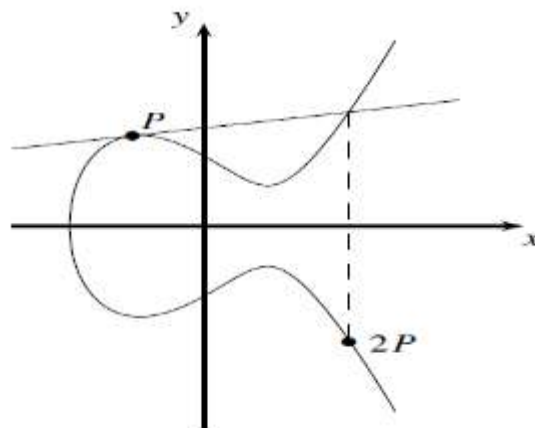


Figure 2.3: Point doubling on an elliptic curve over the real numbers

Let point $\mathbf{P} = (x_1, y_1)$ exist on the curve where $x_1 \neq 0$. The doubling of the point, $2\mathbf{P} = \mathbf{R}$ we can write $\mathbf{R} = \mathbf{P} + \mathbf{P} = 2\mathbf{P}$.

In a cryptosystem we cannot perform geometric constructions. However, by applying simple coordinate geometry, we can express both of the geometric constructions. In particular, we can take the curve equation from equation (2.1), but we now consider it over prime fields $\mathbf{F}_p$ rather than over the real numbers. This yields the following analytical expressions for the group operation.

if $\mathbf{P} \neq \mathbf{Q}$ (point addition)

$$(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$$

$$\lambda = (y_2 - y_1)(x_2 - x_1)^{-1} \bmod p$$

$$x_3 = \lambda^2 - x_1 - x_2 \bmod p \dots\dots\dots(2.3)$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \bmod p \dots\dots\dots(2.4)$$

if $\mathbf{P} = \mathbf{Q}$ (point doubling)

$$(x_1, y_1) + (x_1, y_1) = (x_3, y_3)$$

$$\lambda = (3x_1^2 + a)(2y_1)^{-1} \bmod p$$

$$x_3 = \lambda^2 - 2x_1 \bmod p \dots\dots\dots(2.5)$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \bmod p \dots\dots\dots(2.6)$$

Note that the parameter λ is the slope of the line through $\mathbf{P}$ and $\mathbf{Q}$ in the case of point addition, or the slope of the tangent through $\mathbf{P}$ in the case of point doubling. Even though we made major headway towards the establishment of a finite group, we are not there yet. One thing that is still missing is an identity (or neutral) element $\mathbf{O}$ such that:

$\mathbf{P} + \mathbf{O} = \mathbf{P}$ point at infinity as the neutral element $\mathbf{O}$. This point at infinity can be visualized as a point that is located towards “plus” infinity along the y-axis or towards “minus” infinity along the y-axis. According the group definition, we can now also define the inverse $-\mathbf{P}$ of any group element $\mathbf{P}$ as:

$$\mathbf{P} + (-\mathbf{P}) = \mathbf{O}.$$

The inverse of the point $P = (x_p, y_p)$ is the point $-P = (x_p, -y_p)$, i.e., the point that is reflected along the x-axis. Figure 2.4 shows the point P together with its inverse. Note that finding the inverse of a point $P = (x_p, y_p)$ is now trivial. We simply take the negative of its y coordinate. In the case of elliptic curves over a prime field F_p (the most interesting case in cryptography), this is easily achieved since $-y_p \equiv P - y_p \bmod p$.

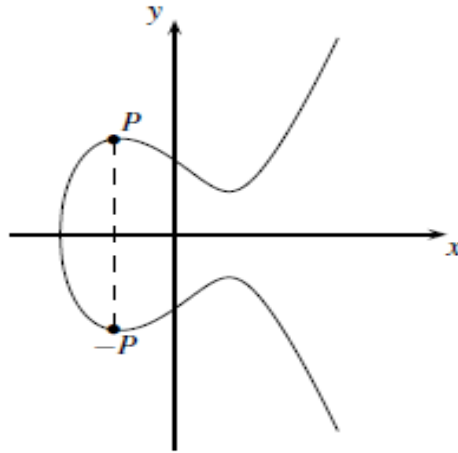


Figure 2.4: The inverse of a point P on an elliptic curve

2.5.3 Elliptic Curved Discrete Logarithm Problem

Definition Elliptic Curved Discrete Logarithm Problem (ECDLP) Given is an elliptic curve E . We consider a primitive element P and another element T . The DL problem is finding the integer d , where $1 \leq d \leq \#E$, such that:

$$P + P + \dots + P = dP = T \dots \dots \dots (2.7)$$

d times where the number of points on the curve is denoted by $\#E$.

In cryptosystems, d is the private key which is an integer, while the public key T is a point on the curve with coordinates $T = (x_T, y_T)$. In contrast, in the case of the DL problem in $\mathbb{Z}_p^*$, both keys were integers. The operation in equation (2.7) is called point multiplication, since we can formally write $T = dP$ [6].

To set up DL cryptosystems it is important to know the order of the group. Even though knowing the exact number of points on a curve is an elaborate task, we know the approximate number due to Hasse's theorem.

Hasse's theorem are: given an elliptic curve E modulo P , the number of points on the

curve is denoted by $\#E$ and is bounded by:

$$P + 1 - 2\sqrt{P} \leq \#E \leq P + 1 + 2\sqrt{P} \dots\dots\dots(2.8)$$

The equation (2.4) is also known as Hasse's bound, states that the number of points is roughly in the range of the prime P . This has major practical implications.

For instance, if we need an elliptic curve with 2^{160} elements, we have to use a prime of length of about 160 bit[6].

2.5.4 Elliptic Curve: Some Definitions

- **Scalar Multiplication:** Given an integer k and a point P on the elliptic curve, the elliptic scalar multiplication kP is the result of adding Point P to itself k times.
- **Curve Order:** The number of points on the elliptic curve is called its curve order and is denoted $\#E$.

2.5.5 Elliptic Curve Cryptography (ECC) domain parameters

The public key cryptographic system involves arithmetic operations on Elliptic curve over finite fields which is determined by elliptic curve domain parameters. The ECC domain parameters over F_p is defined by the given below

$D = (p, a, b, G, n, h)$, where

p: prime number.

a, b: field elements, they specify the equation of the elliptic curve E over F_p ,

$$y^2 = x^3 + a \cdot x + b$$

G: A base point represented by $G = (x_g, y_g)$ on $E(F_p)$.

n: Order of point G , that is n is the smallest positive integer such that $nG = O$.

h: cofactor, and is equal to the ratio $\#E(F_p)/n$, where $\#E(F_p)$ is the curve order.

The primary security in ECC is the parameter n ; therefore the length of ECC key is the bit length of n . For comparative length, the security of ECC keys is much more than that of other cryptosystems. That is for equivalent security, the key length of ECC key is much lesser than other cryptosystems.

2.5.6 Diffie–Hellman Key Exchange with Elliptic Curves

Key exchange using elliptic curves; this is referred to as elliptic curve Diffie–Hellman key exchange, or ECDH. Key exchange using elliptic curves can be done in the following manner. Choose a prime p and the elliptic curve $E : y^2 \equiv x^3 + a \cdot x + b \pmod{p}$. Choose a primitive element $G = (x_G, y_G)$. The prime p , the curve given by its coefficients a, b , and the primitive element G . The actual key exchange is done the same way it was done for the conventional Diffie–Hellman protocol.

A key exchange between users Amina and Bashir can be accomplished as follows:

1. Amina choose private key $a \in \{2, 3, \dots, \#E - 1\}$.
2. Amina compute public key $aG = A = (x_A, y_A)$ and send it to Bashir.
3. Bashir choose private key $b \in \{2, 3, \dots, \#E - 1\}$.
4. Bashir compute public key $bG = B = (x_B, y_B)$ and send it to Amina .
5. Amina compute $aB = T_{AB}$.
6. Bashir compute $bA = T_{AB}$.

In fact both parties compute the same result, namely the point $T_{AB} = abG$.

Example:

Let E be an elliptic curve define over F_p where $p = 3023$ with parameters $a = 1, b = 2547$ where $(4a^3 + 27b^2) \pmod{p} = 2027 \neq 0$. and $\#E = 3083$. Therefore let $G = (2237, 2480)$ be a point on E .

Alice chooses a secret random integer $e = 2313$.

$$eB = 2313 (2237, 2480) = (934, 29)$$

And send (934, 29) to Bob .

- Bob chooses a secret random integer $d = 1236$.

$$dB = 1236 (2237, 2480) = (1713, 1709)$$

And send (1713, 1709) to Alice

- Alice computes the secret key $e (dB) = 2313 (1713, 1709)$.

$$edB = (2537, 1632) = K$$

- Bob computes the secret key $d (eB) = 1236 (934, 29)$.

$$deB = (2537, 1632) = K$$

Now, Alice and Bob have the same point $K = (2537, 1632)$.

2.5.7 Security of Elliptic Curve Cryptography

The security of ECC depends on how difficult it is to determine k given kP and P . This is referred to as the elliptic curve logarithm problem [2].

All three of the established public-key algorithm families (Integer-Factorization Schemes, Discrete Logarithm Schemes and Elliptic Curve Schemes) are based on number theoretic functions. One distinguishing feature of them is that they require arithmetic with very long operands and keys. Not surprisingly, the longer the operands and keys, the more secure the algorithms become. In order to compare different algorithms, one often considers the security level. An algorithm is said to have a “security level of n bit” if the best known attack requires 2^n steps. This is a quite natural definition because symmetric algorithms with a security level of n have a key of length n bit. The relationship between cryptographic strength and security is not as straightforward in the asymmetric case, though. Table 2.2 shows recommended bit lengths for public-key algorithms for the four security levels 80, 128, 192 and 256 bit. We see from the table that RSA-like schemes and discrete-logarithm schemes require very long operands and keys. The key length of elliptic curve schemes is significantly smaller, yet still twice as long as symmetric ciphers with the same cryptographic strength [6].

Table (2.2) Comparable Key Sizes in Terms of Computational Effort for Cryptanalysis

Algorithm Family	Cryptosystems	Security Level (bit)			
		80	128	192	256
Integer factorization	RSA	1024 bit	3072 bit	7680 bit	15360 bit
Discrete logarithm	DH, DSA, Elgamal	1024 bit	3072 bit	7680 bit	15360 bit
Elliptic curves	ECDH, ECDSA	160 bit	256 bit	384 bit	512 bit
Symmetric-key	AES, Triple-DES	80 bit	128 bit	192 bit	256 bit

Chapter Three
Proposed Solution

3. Elliptic Curve Cryptosystems

This chapter is intended to describe few crypto-systems which are based on elliptic curve and some related problems. And also, the proposed encryption systems.

Elliptic Curve Cryptography uses a group of points (instead of integers) for cryptographic schemes with coefficient sizes of 160-256 bits, reducing significantly the computational effort.

Elliptic curve cryptosystem is a generic term for any cryptosystem that works in the domain of elliptic curves. The first encryption schemes based on elliptic curves were the equivalent versions of the Massey-Omura [8] and ElGamal [9] cryptosystems, both presented by Koblitz in 1985 (and published in 1987) [10], and the Menezes-Vanstone cryptosystem [11].

3.1 Massey-Omura Elliptic Curve Cryptosystem

Massey–Omura cryptographic scheme is a three-pass protocol for sending messages, allowing Amina to securely send a message to Bashir without the need to exchange or distribute encryption keys [12].

We will describe the Massey-Omura protocol by which Amina sends the message M to Bashir, where E is an elliptic curve defined over the finite field F_p , $N = \#E$, and there exists a publicly known relationship between the plaintexts and some points of the curve, so for any message M the point $P_M \in E$ is known by all the parties.

The embedding system $M \rightarrow P_M$, as well as p , E , and N , are publicly known. When Amina wants to communicate secretly with Bashir, then the procedure should be as follows:

Amina chooses a secret pair of numbers (e_A, d_A) such that $e_A * d_A = 1 \pmod{N}$.

1. Similarly, Bashir chooses (e_B, d_B) such that $e_B * d_B = 1 \pmod{N}$.

If Amina wants to send a secret message-point $P_M \in E$ to Bashir, then the procedure should be as follows:

3. Amina sends $e_A P_M \pmod{p}$ to Bashir,
4. Bashir sends $e_B e_A P_M \pmod{p}$ to Amina,
5. Amina sends $d_A e_B e_A P_M \pmod{p} = e_B P_M$ to Bashir,
6. Bashir computes $d_B e_B P_M = P_M$ and hence recovers the original message point.

Example:

$$p = 13 ,$$

$$F_{13}: y^2 = x^3 + 2x + 2 \pmod{17}$$

$$N = 19.$$

$$P_M = (5, 1) ,$$

To finding the points on an elliptic curve and the point at infinity O , can be using equations (2.3), (2.4) if $P \neq Q$ (point addition) and can be using equations (2.5), (2.6) if $P = Q$ (point doubling), as shown in Table 3.1 below, where this elliptic curve has order $\#E = 19$ since it contains 19 points in its cyclic group.

Table (3.1) The points on $y^2 = x^3 + 2x + 2 \pmod{17}$

$2P_M$	(6,3)	$11P_M$	(13,10)
$3P_M$	(10,6)	$12P_M$	(0,11)
$4P_M$	(3,1)	$13P_M$	(16,4)
$5P_M$	(9,16)	$14P_M$	(9,1)
$6P_M$	(16,13)	$15P_M$	(3,16)
$7P_M$	(0,6)	$16P_M$	(10,11)
$8P_M$	(13,7)	$17P_M$	(6,14)
$9P_M$	(7,6)	$18P_M$	(5,16)
$10P_M$	(7,11)	$19P_M$	O

Amina chooses a secret pair of numbers $(e_A, d_A) = (2, 10) \pmod{19}$,

Bashir chooses $(e_B, d_B) = (4, 5) \pmod{19}$.

Then

Amina sends:

$$\begin{aligned} e_AP_M &= 2(5, 1) \pmod{17} , \\ &= (6, 3) \pmod{17} , \end{aligned}$$

Bashir sends:

$$\begin{aligned} e_B e_AP_M &= e_B(6, 3) \pmod{17} , \\ &= 4(6, 3) \pmod{17} , \\ &= (13, 7) \pmod{17} , \end{aligned}$$

Amina sends:

$$d_A e_B e_AP_M = d_A(13, 7) \pmod{17} ,$$

$$= 10 (13, 7) \pmod{17},$$

$$= (3, 1) \pmod{17},$$

Bashir computes:

$$d_B d_A e_B e_A P_M = d_B (3, 1) \pmod{17},$$

$$= 5 (3, 1) \pmod{17},$$

$$= (5, 1) \pmod{17},$$

$$= P_M.$$

3.2 ElGamal Elliptic Curve Cryptography

Just the same as many other public-key cryptosystems, the famous ElGamal cryptosystem also has a very straightforward elliptic curve analog, which may be described as follows:

Amina and Bashir publicly choose an elliptic curve E over F_p and a random base point $G \in E$. Suppose they also know the number of points on E , that is, they know $N = \#E$.

Suppose Bashir wishes to send a secret message P_M to Amina:

Amina chooses a random integer a , computes $aG \pmod{p}$ and sends it to Bashir.

Bashir chooses at random an integer b and computes $bG \pmod{p}$.

Bashir also computes $(P_M + b(aG)) \pmod{p}$. Then Bashir sends the secret encrypted message $(bG, P_M + b(aG)) \pmod{p}$ to Amina.

Amina has the secret key a , she can compute $a(bG) \pmod{p}$ and get

$$P_M = (P_M + a(bG) - b(aG)) \pmod{p}.$$

Where P_M is the original plaintext message.

The eavesdropper can only get P_M if she can solve the Elliptic Curve Discrete Logarithm Problem. That is, she can get P_M if she can find a from $aG \pmod{p}$ or b from $bG \pmod{p}$. But, as everybody knows, there is no efficient way to compute the elliptic curve discrete logarithms [12].

Example: Suppose Bashir wishes to send Amina a secret message P_M by using the elliptic curve ElGamal cryptographic scheme.

$$F_{29}: y^2 = x^3 - x + 16 \pmod{29}$$

$$N = 31.$$

$$G = (5, 7) \in E(F_{29}),$$

$$P_M = (28, 25),$$

Public-key generation: Assume Bashir sends the secret message P_M to Amina, so Amina:

Chooses a random secret integer $a = 23$,

Computes $aG = 23G = (21, 18) \pmod{29}$,

Sends $aG = (21, 18) \pmod{29}$ to Bashir.

Encryption: Bashir

Chooses a random secret integer $b = 25$,

Computes $bG = 25G = (13, 24) \pmod{29}$,

$b(aG) = 25(23G) = 25(21, 18) = (1, 25) \pmod{29}$,

$P_M + b(aG) = (28, 25) + (1, 25) = (0, 4) \pmod{29}$,

Sends $(bG = (13, 24), P_M + b(aG) = (0, 4))$ to Amina.

Decryption: Amina computes

$a(bG) = 23(25G) = 23(13, 24) = (1, 25)$,

$P_M = P_M + b(aG) - a(bG)$

$= (0, 4) - (1, 25)$

$= (0, 4) + (1, -25)$

$= (28, 25)$.

So, Amina recovers the original secret message $P_M = (28, 25)$.

3.3 Menezes–Vanstone Elliptic Curve Cryptography

A serious problem with Massey-Omura and ElGamal elliptic curve cryptosystems is that the plaintext message units P_M lie on the elliptic curve E , and there is no convenient method known of deterministically generating such points on E [12].

The solution to the problem of encoding a message in a point is the Menezes-Vanstone Elliptic Curve Cryptosystem (MVECC). In this variation (MVECC), the EC is used for “masking”, and plaintexts and ciphertexts are allowed to be arbitrary ordered pairs of (nonzero) elements (i.e., they are not required to be points on E) [13], [14]. Fortunately, Menezes and Vanstone discovered a more efficient variation in this variation which we shall describe below, the elliptic curve is used for “masking” and the plaintext and ciphertext pairs are allowed to be in $F_p^* \times F_p^*$ rather than on the elliptic curve.

If Amina wants to send a message to Bashir using his public key. Amina and Bashir have to decide upon the following conventions, all of which are public.

- p a large prime number (it must at least be larger than 3).
- F_p a finite field of size p (p is prime, so it works like modular arithmetic).

- E an elliptic curve over F_p of the form $y^2 = x^3 + ax + b, (a, b \text{ in } F_p)$.
- G a randomly selected point on E (the base point) that will generate subgroups.

Each user chooses a random integer d_x which will be his/her own secret key, then computes and publishes the point $d_x G$.

Suppose Amina wishes to send a message $M = (m_1, m_2) \in F_p^* \times F_p^*$ to Bashir.

Bashir's private key. Only he knows it, Bashir's private key is a randomly selected $d_b \in \#E$.

Public Key: Bashir's public key. Ideally it is distributed to the world.

B : Bashir's public key is calculated as $B = d_b G$.

Encryption:

Amina has secret M , which she splits up into m_1 and m_2 ,

calculates $(k_1, k_2) = d_a B$.

calculates $C_0 = d_a G$. Note that C_0 is a point.

calculates $C_1 = k_1 m_1 \text{ mod } p$.

calculates $C_2 = k_2 m_2 \text{ mod } p$.

Amina sends encrypted message $C = (C_0, C_1, C_2)$ to Bashir.

Decryption:

Bashir wants to get back the message M from C .

Bashir calculates $d_b C_0 = (k_1, k_2)$,

retrieves message M by calculating

$$m_1 = C_1 k_1^{-1} \text{ mod } p.$$

$$m_2 = C_2 k_2^{-1} \text{ mod } p.$$

$$M = (m_1, m_2).$$

Example:

Let E be an elliptic curve define over F_p where $p = 3023$ with parameters $a = 1, b = 2547$ where $(4a^3 + 27b^2) \text{ mod } p = 2027 \neq 0$. and $\#E = 3083$. Therefore let $G = (2237, 2480)$ be a point on E .

Bashir chooses a secret random integer $d_b = 2313$. It is Bashir's private key.

Bashir's public key is $B = d_b G = 2313(2237, 2480) = (934, 29)$, and the plaintext $M = (1234, 1423) = (m_1, m_2)$.

Encryption:

Encryption: Amina computes:

$$(k_1, k_2) = d_a B = 1236(934, 29) = (2537, 1632),$$

$$C_0 = d_a G = 1236(2237, 2480) = (1713, 1709),$$

$$C_1 = k_1 m_1 = (2537 * 1234) \bmod 3023 = 1853,$$

$$C_2 = k_2 m_2 = (1632 * 1423) \bmod 3023 = 672.$$

Then Amina sends

$$C = (C_0, C_1, C_2) = ((1713, 1709), 1853, 672) \text{ to Bashir.}$$

Decryption: Upon receiving Amina's message, Bashir computes:

$$d_b C_0 = 2313(1713, 1709) = (2537, 1632) = (k_1, k_2),$$

$$m_1 = C_1 k_1^{-1} \bmod p = 1234,$$

$$m_2 = C_2 k_2^{-1} \bmod p = 1423.$$

Thus, Bashir recovers the message $M = (1234, 1423)$.

3.4 Al-Saffar and Said's Algorithm

In 2013 a research [3] by Al-Saffar and Said introduced proposed to development of MVECC this system which the development of the encryption and decryption of MVECC are as follows:

Suppose Amina wants to send a message $M \in F_p^* \times F_p^*$ to Bashir.

Key generation:

Bashir's private key. Only he knows it, Bashir's private key is a randomly selected $d_b \in \#E$.

Public Key: Bashir's public key. Ideally it is sent to Amina.

B: Bashir's public key is calculated as $B = d_b G$.

Encryption:

Amina has message $M = (m_1, m_2) \in F_p^* \times F_p^*$,

calculates $(k_1, k_2) = d_a B$.

calculates $C_0 = d_a G$. Note that C_0 is a point.

calculates $C_1 = m_1 + k_1 k_2 \bmod p$.

calculates $C_2 = m_1(m_2 + k_2 k_1) \bmod p$.

Amina sends encrypted message $C = (C_0, C_1, C_2)$ to Bashir.

Decryption:

Bashir wants to get back the message M from C .

Bashir calculates $d_b C_0 = (k_1, k_2)$,

retrieves message M by calculating

$$m_1 = C_1 - k_1 k_2 \bmod p,$$

$$m_2 = m_1^{-1} C_2 - k_1 k_2 \bmod p.$$

Thus, Bashir recovers the message M

Where $M = (m_1, m_2)$.

Example:

Let E be an elliptic curve define over F_p where $p = 3023$ with parameters $a = 1, b = 2547$ where $(4a^3 + 27b^2) \bmod p = 2027 \neq 0$. and $\#E = 3083$. Therefore let $G = (2237, 2480)$ be a point on E .

Bashir chooses a secret random integer $d_b = 2313$. It is Bashir's private key.

Bashir's public key is $B = d_b G = 2313(2237, 2480) = (934, 29)$.

Encryption:

Amina represent the message $M = (1111, 2222) = (m_1, m_2)$.

Chooses a secret random integer $d_a = 1236$.

Amina computes:

$$(k_1, k_2) = d_a B = 1236(934, 29) = (2537, 1632),$$

$$C_0 = d_a G = 1236(2237, 2480) = (1713, 1709),$$

calculates $C_1 = m_1 + k_1 k_2 \bmod p$

$$= 1111 + 2537 * 1632 \bmod 3023$$

$$= 3008.$$

calculates $C_2 = m_2(m_2 + k_2 k_1) \bmod p$

$$= 1111(2222 + 1632 * 2537) \bmod 3023$$

$$= 2410.$$

Amina sends encrypted message $C = ((1713, 1709), 3008, 2410)$ to Bashir.

Decryption:

Bashir computes:

$$d_b C_0 = 2313(1713, 1709) = (2537, 1632) = (k_1, k_2),$$

$$m_1 = C_1 - k_1 k_2 \bmod p$$

$$\begin{aligned}
 &= 3008 - 2537 * 1632 \bmod 3023 \\
 &= 1111, \\
 m_2 &= m_1^{-1}C_2 - k_1k_2 \bmod p. \\
 &= 1453 * 2410 - 2537 * 1632 \bmod 3023 \\
 &= 2222.
 \end{aligned}$$

Thus, Bashir recovers the message $M = (1111, 2222)$.

3.5 Proposed Methods of MVECC

We suggest two cryptographic methods that modify MVECC algorithm. In the elliptic curve encryption/decryption schemes, the plaintext becomes twice as long as the cipher text. This characteristic may be exploited to make the system more efficient in performance. The Menezes and Vanstone introduced the variant of ElGamal analogue. This is the implementation of ECC without needing to embed the plaintext unit in the curve (i.e. represent the plaintext unit as any two integer number in the field, accept in the curve equation or not). This is the major difference from the others ECC specially ElGamal cryptosystem, since there is a need to embed the plaintext unit in the curve by representing the plaintext unit as an acceptable point in the curve. But, Menezes-Vanstone algorithm needs to compute the inverses of each number in the key point. And the inverse operation is an expansive operation.

If we see the encryption function in the Menezes-Vanstone algorithm, it computes two numbers (i.e. ciphertext unit (C_1, C_2)) from four numbers, two of these are the plaintext (m_1, m_2) and the other two numbers are the key point (k_1, k_2) .

To restore the plaintext the decryption function computes the k_1^{-1} and k_2^{-1} it is clear that finding the plaintext requires two inverse operations.

In our proposal we attempt to avoid the inverse operation, not as in the algorithm MVECC we are trying to avoid the process of inverse when decrypt. Therefore we propose the following algorithms.

Algorithm I:

Suppose Amina wants to send a message $M \in F_p^* \times F_p^*$ to Bashir.

Key generation:

Bashir's private key. Only he knows it, Bashir's private key is a randomly selected $d_b \in \#E$.

Public Key: Bashir's public key. Ideally it is sent to Amina.

B : Bashir's public key is calculated as $B = d_b G$.

Encryption:

Amina has message $M = (m_1, m_2) \in F_p^* \times F_p^*$,

calculates $(k_1, k_2) = d_a B$.

calculates $C_0 = d_a G$. Note that C_0 is a point.

calculates $C_1 = m_1 k_1 - m_2 \bmod p$.

calculates $C_2 = k_2(m_2 - m_1 k_1) + m_1 \bmod p$.

Amina sends encrypted message $C = (C_0, C_1, C_2)$ to Bashir.

Decryption:

Bashir wants to get back the message M from C .

Bashir calculates $d_b C_0 = (k_1, k_2)$,

retrieves message M by calculating

$$m_1 = C_2 + k_2 C_1 \bmod p,$$

$$m_2 = m_1 k_1 - C_1 \bmod p.$$

Thus, Bashir recovers the message M

Where $M = (m_1, m_2)$.

Example:

Let E be an elliptic curve define over F_p where $p = 3023$ with parameters $a = 1, b = 2547$ where $(4a^3 + 27b^2) \bmod p = 2027 \neq 0$. and $\#E = 3083$. Therefore let $G = (2237, 2480)$ be a point on E .

Bashir chooses a secret random integer $d_b = 2313$. It is Bashir's private key.

Bashir's public key is $B = d_b G = 2313(2237, 2480) = (934, 29)$.

Encryption:

Amina represent the message $M = (1111, 2222) = (m_1, m_2)$.

Chooses a secret random integer $d_a = 1236$.

Amina computes:

$$(k_1, k_2) = d_a B = 1236(934, 29) = (2537, 1632),$$

$$C_0 = d_a G = 1236(2237, 2480) = (1713, 1709),$$

$$\begin{aligned}\text{calculates } C_1 &= m_1 k_1 - m_2 \bmod p \\ &= 1111 * 2537 - 2222 \bmod 3023 \\ &= 1972.\end{aligned}$$

$$\begin{aligned}\text{calculates } C_2 &= k_2(m_2 - m_1 k_1) + m_1 \bmod p \\ &= 1632 (2222 + 1111 * 2537) + 1111 \bmod 3023 \\ &= 2302.\end{aligned}$$

Amina sends encrypted message $C = ((1713, 1709), 1972, 2302)$ to Bashir.

Decryption:

Bashir computes:

$$d_b C_0 = 2313(1713, 1709) = (2537, 1632) = (k_1, k_2),$$

$$\begin{aligned}m_1 &= C_2 + k_2 C_1 \bmod p \\ &= 2302 + 1632 * 1972 \bmod 3023 \\ &= 1111 ,\end{aligned}$$

$$\begin{aligned}m_2 &= m_1 k_1 - C_1 \bmod p \\ &= 1111 * 2537 - 1972 \bmod 3023 \\ &= 2222 .\end{aligned}$$

Thus, Bashir recovers the message $M = (1111, 2222)$.

Algorithm II:

Suppose Amina wants to send a message $M \in F_p^* \times F_p^*$ to Bashir.

Key generation:

Bashir's private key. Only he knows it, Bashir's private key is a randomly selected $d_b \in \#E$.

Public Key: Bashir's public key. Ideally it is sent to Amina.

B : Bashir's public key is calculated as $B = d_b G$.

Encryption:

Amina has message $M = (m_1, m_2) \in F_p^* \times F_p^*$,

calculates $(k_1, k_2) = d_a B$.

calculates $C_0 = d_a G$. Note that C_0 is a point.

calculates $C_1 = m_2 k_2 - m_1 \bmod p$.

calculates $C_2 = k_1(m_1 - m_2 k_2) + m_2 \bmod p$.

Amina sends encrypted message $C = (C_0, C_1, C_2)$ to Bashir.

Decryption:

Bashir wants to get back the message M from C .

Bashir calculates $d_b C_0 = (k_1, k_2)$,

retrieves message M by calculating

$$m_2 = C_2 + k_1 C_1 \bmod p,$$

$$m_1 = m_2 k_2 - C_1 \bmod p.$$

Thus, Bashir recovers the message M

Where $M = (m_1, m_2)$.

Example:

Let E be an elliptic curve define over F_p where $p = 3023$ with parameters $a = 1, b = 2547$ where $(4a^3 + 27b^2) \bmod p = 2027 \neq 0$. and $\#E = 3083$. Therefore let $G = (2237, 2480)$ be a point on E .

Bashir chooses a secret random integer $d_b = 2313$. It is Bashir's private key.

Bashir's public key is $B = d_b G = 2313(2237, 2480) = (934, 29)$.

Encryption:

Amina represent the message $M = (1111, 2222) = (m_1, m_2)$.

Chooses a secret random integer $d_a = 1236$.

Amina computes:

$$(k_1, k_2) = d_a B = 1236(934, 29) = (2537, 1632),$$

$$C_0 = d_a G = 1236(2237, 2480) = (1713, 1709),$$

$$\begin{aligned} \text{calculates } C_1 &= m_2 k_2 - m_1 \bmod p \\ &= 2222 * 1632 - 1111 \bmod 3023 \\ &= 616. \end{aligned}$$

$$\begin{aligned} \text{calculates } C_2 &= k_1(m_1 - m_2 k_2) + m_2 \bmod p \\ &= 2537(1111 + 2222 * 1632) + 2222 \bmod 3023 \\ &= 2321. \end{aligned}$$

Amina sends encrypted message $C = ((1713, 1709), 616, 2321)$ to Bashir.

Decryption:

Bashir computes:

$$d_b C_0 = 2313(1713, 1709) = (2537, 1632) = (k_1, k_2),$$

$$\begin{aligned} m_2 &= C_2 + k_1 C_1 \\ &= 2321 + 2537 * 616 \text{ mod } 3023 \\ &= 2222 , \end{aligned}$$

$$\begin{aligned} m_1 &= m_2 k_2 - C_1 \text{ mod } p \\ &= 2222 * 1632 - 616 \text{ mod } 3023 \\ &= 1111 . \end{aligned}$$

Thus, Bashir recovers the message $M = (1111, 2222)$.

Chapter Four

Results and Discussion

4.1 Experimental Setup

The simulation software was written in Java programming language. As Java allows the selection amongst five different curves defined over $F(p)$. The user has the option of selecting one of the following National Institute of Standards and Technology (NIST) curves via a dropdown list: **P-192**, **P-224**, **P-256**, **P-384**, and **P-512**. The names of these curves reference the bit length of the keys and are listed by increasing strength. NIST curves are used because they are recommended as a secure standard under Federal Information Processing Standards (FIPS 186-2).

The following packages and classes were used in the simulation:

- `java.io.*`; to initialize files using (`File`, `FileInputStream`, `FileOutputStream`) classes.
- `java.io.FileInputStream`; a `FileInputStream` obtains input bytes from a file in a file system.
- `java.io.FileOutputStream`; is an output stream for writing data to a `File`.
- `java.io.IOException`; is the general class of exceptions produced by failed or interrupted I/O operations.
- `java.math.BigInteger`; this class work with integers that are larger than 64 bits; and It also provides operations for modular arithmetic.
- `java.security.SecureRandom`; `SecureRandom` class is used to generate a cryptographically strong pseudo random number.
- `org.bouncycastle.asn1.nist.NISTNamedCurves`; class for fetching curves using their NIST names.
- `org.bouncycastle.math.ec.ECPoint`; base class for points on elliptic curves.
- `org.bouncycastle.asn1.x9.X9ECParameters`; class for Elliptic-Curve EC Parameters structure.

To evaluate and compare the performance of the MVECC system with the proposed algorithms, two keys k_1, k_2 were used for encrypting the data. And then assess independently each of the two main components (encryption and decryption) based on running times. Basically, the fundamental parameters of those operations are:

- a. The size of the applied key and,
- b. The size and content of the input data.

All tests were performed on Intel Core(TM) i5-3210M 2.50GHz with 6144MB of RAM. The tests used curve P-256 which has 256 bits keys. Also, the proposed algorithms used it with text files having different sizes. When it takes a plaintext file of 2 KB size then encrypts it and computes the execution time of this operations. Then decrypts it and again records the execution time of such operations. Then this process is repeated for 4 KB, 6 KB, 8 KB, 16 KB and 32 KB text files. And the running times of the encryption and decryption operations of each text file are stored.

4.2 Obtained Results

In the following sections the results obtained during experiments will be presented. That is results for MVECC, algorithms, Al-Saffar and Said's algorithm cryptosystem and the proposed algorithms. The computational complexity of the encryption and decryption functions for MVECC algorithm and our proposal will be computed. The results obtained from these experiments are analyzed, interpreted and graphically presented. Specifically, it shows the execution time of each algorithm for the sake of performance evaluation for encryption and decryption as shown in Table 4.1, Table 4.2 and Table 4.3 below. Where the time required by an algorithm to complete the operation depends on processor speed and algorithm complexity. Shorter the time the algorithm takes to complete its operation better it is.

Table (4.1): Execution Time (Milliseconds) of Encryption of different text file size

Algorithm	2KB	4KB	6KB	8KB	16KB	32KB
MVECC	0.60	0.99	1.30	1.65	2.88	5.20
Al-Saffar and Said's algorithm	0.64	1.02	1.41	1.74	3.08	5.67
Algorithm I	0.64	1.02	1.44	1.78	3.11	5.80
Algorithm II	0.64	1.02	1.44	1.78	3.11	5.80

Table (4.2): Execution Time (Milliseconds) of Decryption of different text file size

Algorithm	2KB	4KB	6KB	8KB	16KB	32KB
MVECC	1.80	3.14	4.33	5.60	11.00	20.40
Al-Saffar and Said's algorithm	1.24	2.09	3.07	3.85	7.34	14.30
Algorithm I	0.60	0.92	1.25	1.58	2.88	5.20
Algorithm II	0.60	0.92	1.25	1.58	2.88	5.20

Table (4.3): Execution Time (Milliseconds) of Encryption and Decryption of different text file size

Algorithm	2KB	4KB	6KB	8KB	16KB	32KB
MVECC	2.40	4.13	5.63	7.25	13.88	25.60
Al-Saffar and Said's algorithm	1.88	3.11	4.48	5.59	10.42	19.97
Algorithm I	1.24	1.94	2.69	3.36	5.99	11.00
Algorithm II	1.24	1.94	2.69	3.36	5.99	11.00

4.3 Analysis and Discussions

The development of elliptic curve cryptosystems based on the mathematical framework of complexity theory and the time which is taken to implement them. In this section all the algorithms proposed in chapter three will be analyzed according to their mathematical complexity and the time of implementation.

4.3.1 Computational Complexity

In this section, the computational complexity of the encryption and decryption functions for the Menezes-Vanstone Elliptic Curve Cryptosystem, Al-Saffar and Said's algorithm and the two proposed methods are calculated. The O -notation has been extremely useful in helping analyst to classify algorithms by performance and in guiding algorithm designers to search for the "best" algorithms for important problem. So adding two s -bit numbers requires s bit operations. That is [15]:

$$T(s\text{-bit} + s\text{-bit}) = O(s)$$

For the input numbers of size n decimal digits

$$T(n + n) = O(\log n), \text{ where the number of bits of } n \text{ equal } \log_2 n.$$

The multiplication of two s -bits binary integers requires s^2 ($s * s$) bit operation, because it needs s addition operations. That is:

$$T(s\text{-bit} * s\text{-bit}) = O(s^2)$$

For the input numbers of size n decimal digits

$$T(n * n) = O(\log n)^2, \text{ where the number of bits of } n \text{ equals } \log_2 n.$$

The Computational Complexity for the MVECC compared to the proposed methods is as follows:

- **Menezes-Vanstone Complexity**

Let the size of the input message unit is n in MVECC Method the encryption function is :

$$C_1 = k_1 * m_1$$

$$C_2 = k_2 * m_2 \text{ then:}$$

$$T(C_1) = O(\log n)^2 \text{ bit operation.}$$

$$T(C_2) = O(\log n)^2 \text{ bit operation.}$$

The decryption function is:

$$m_1 = C_1 * k_1^{-1} \text{ mod } p$$

$$m_2 = C_2 * k_2^{-1} \text{ mod } p \text{ then:}$$

$$T(m_1) = O(\log n)^2 + T(k_1^{-1})$$

$$T(k_1^{-1}) = O(\log n)^3, \text{ by extended Euclid's method.}$$

$$T(m_1) = O(\log n)^2 + O(\log n)^3 \text{ bit operation.}$$

$$T(m_2) = O(\log n)^2 + O(\log n)^3 \text{ bit operation.}$$

- **Al-Saffar and Said's algorithm Complexity**

Let the size of the input message unit is n in Al-Saffar and Said Algorithm Method the encryption function is :

$$C_1 = m_1 + k_1 k_2 \text{ mod } p.$$

$$C_2 = m_1(m_2 + k_2 k_1) \text{ mod } p.$$

$$T(C_1) = O(\log n) + O(\log n)^2 \text{ bit operation.}$$

$$T(C_2) = O(2(\log n)^2) + O(\log n) \text{ bit operation.}$$

The decryption function is:

$$m_1 = C_1 - k_1 k_2 \text{ mod } p,$$

$$m_2 = m_1^{-1}C_2 - k_1k_2 \bmod p.$$

$$T(m_1) = O(\log n) + O(\log n)^2 \text{ bit operation.}$$

$$T(m_2) = T(m_1^{-1}) + O(2(\log n)^2) + O(\log n) \text{ bit operation.}$$

$$T(m_1^{-1}) = O(\log n)^3, \text{ by extended Euclid's method.}$$

$$T(m_2) = O(\log n)^3 + O(2(\log n)^2) + O(\log n).$$

- **Algorithm I Complexity**

Let the size of the input message unit is n in Algorithm I Method the encryption function is :

$$C_1 = m_1 k_1 - m_2 \bmod p.$$

$$C_2 = k_2(m_2 - m_1 k_1) + m_1 \bmod p \text{ then:}$$

$$T(C_1) = O(\log n)^2 + O(\log n) \text{ bit operation.}$$

$$T(C_2) = O(2(\log n)^2) + O(2 \log n) \text{ bit operation.}$$

The decryption function is:

$$m_1 = C_2 + k_2 C_1 \bmod p,$$

$$m_2 = m_1 k_1 - C_1 \bmod p \text{ then:}$$

$$T(m_1) = O(\log n) + O(\log n)^2 \text{ bit operations.}$$

$$T(m_2) = O(\log n)^2 + O(\log n) \text{ bit operations.}$$

- **Algorithm II Complexity**

Let the size of the input message unit is n in Algorithm II Method the encryption function is :

$$C_1 = m_2 k_2 - m_1 \bmod p.$$

$$C_2 = k_1(m_1 - m_2 k_2) + m_2 \bmod p \text{ then:}$$

$$T(C_1) = O(\log n)^2 + O(\log n) \text{ bit operation.}$$

$$T(C_2) = O(2(\log n)^2) + O(2 \log n) \text{ bit operation.}$$

The decryption function is:

$$m_2 = C_2 + k_1 C_1 \text{ mod } p,$$

$$m_1 = m_2 k_2 - C_1 \text{ mod } p \text{ then:}$$

$$T(m_2) = O(\log n) + O(\log n)^2 \text{ bit operations.}$$

$$T(m_1) = O(\log n)^2 + O(\log n) \text{ bit operations.}$$

The mathematical complexity of any elliptic curve cryptosystem depends on the largest prime factor of the group order of the elliptic which is used by the system. And on the number of operations which are used during the encryption and decryption processes algorithms [3].

So, we can discuss the mathematical complexity according to the above fact as follows:

The Al-Saffar and Said's algorithm of Menezes-Vanstone elliptic curve cryptosystem is more efficient than the Menezes-Vanstone elliptic curve cryptosystem, where in the encryption scheme there are three multiplication operations ($k_1 k_2$, $m_1 m_2$ and $m_1 k_2 k_1$) and two addition operations. While the decryption scheme needed to compute the inverse operations for m_1 , and three multiplication operations ($k_1 k_2$, $m_1^{-1} C_2$ and $m_1^{-1} k_1 k_2$) and two subtraction operations.

The proposed Algorithm I of Menezes-Vanstone elliptic curve cryptosystem is more efficient than the MVECC and previous proposition (Al-Saffar and Said's algorithm), where in the encryption scheme there are three multiplication operations ($m_1 k_1$, $k_2 m_2$, $k_2 m_1 k_1$) and two subtraction operations and one addition operation. While the decryption scheme needed to compute two multiplication operations, one addition operation and one subtraction operation.

The proposed Algorithm II of Menezes-Vanstone elliptic curve cryptosystem is more efficient than the MVECC and previous proposition (Al-Saffar and Said's algorithm), where in the encryption scheme there are three multiplication operations ($m_2 k_2$, $k_1 m_1$ and $k_1 m_2 k_2$), and two subtraction operations also one addition operation. While the decryption scheme needed to compute the two multiplication operations ($k_1 C_1$ and $m_2 k_2$), one addition operation and one subtraction operation.

4.3.2 Execution Time for Encryption

Execution time of encrypting different text files using the MVECC, (Al-Saffar and Said's algorithm) and two proposed algorithms is shown in Figure 4.1.

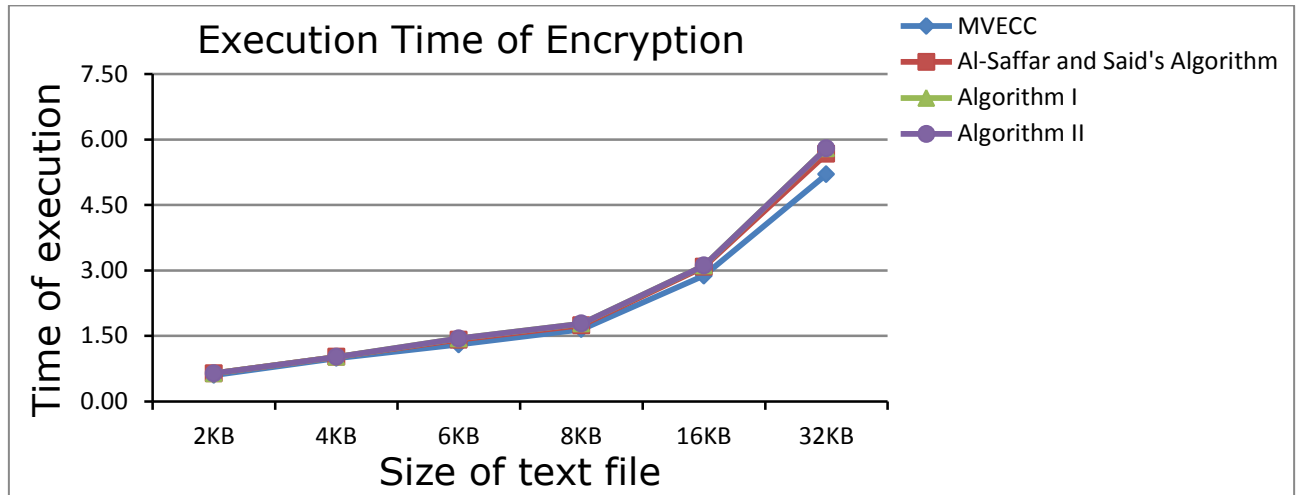


Figure 4.1: Execution time for encryption (in milliseconds).

Figure 4.1 shows the result based on the execution time of the encryption with different size of text files. It shows that the execution time is approximate equivalent for all proposed algorithms compared to that of MVECC algorithm.

4.3.3 Execution Time for Decryption

Execution time for decryption to the MVECC, (Al-Saffar and Said's algorithm) and two proposed algorithms is shown in Figure 4.2

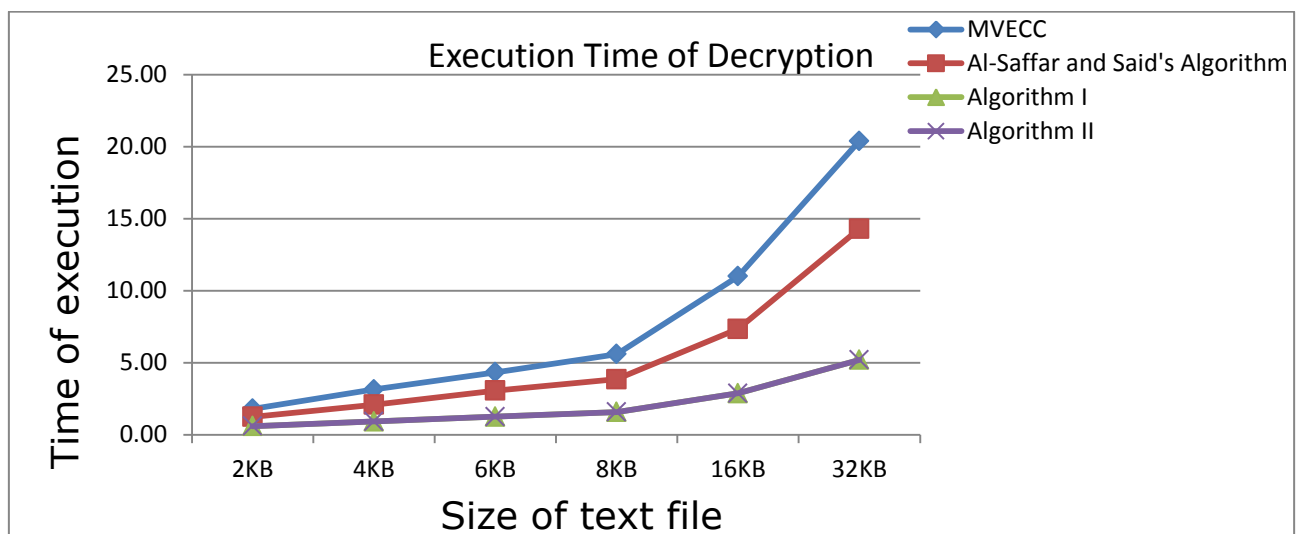


Figure 4.2: Execution time for decryption (in milliseconds)

Figure 4.2 shows the result based on the execution time of the decryption with different size of text files. It shows that the (Al-Saffar and Said's algorithm) take less execution time from MVECC algorithm, Algorithm I and Algorithm II take less execution time from the (Al-Saffar and Said's algorithm) and MVECC algorithm.

Then execution time is minimum for Algorithm I and Algorithm II when compared to that of (Al-Saffar and Said's algorithm) and MVECC algorithm.

4.3.4 Execution Time for Encryption and Decryption

Execution time for encryption and decryption of MVECC, (Al-Saffar and Said's algorithm) and two proposed algorithms is shown in Figure 4.3

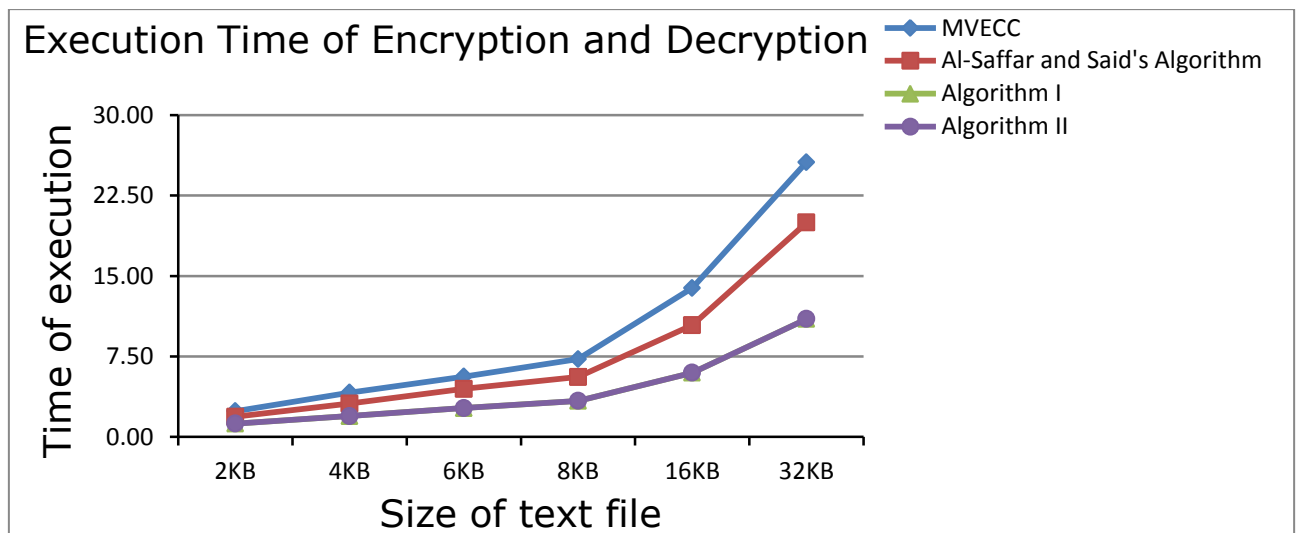


Figure 4.3: Execution time for encryption and decryption (in milliseconds).

Figure 4.3 shows the result based on the total execution time for encryption and decryption with different size of text files. It shows that the execution time is minimum for Algorithm I, Algorithm II when compared to that of (Al-Saffar and Said's algorithm) and MVECC algorithm.

Algorithm I and Algorithm II are two modified versions of the MVECC, each variation guarantees an improved performance in terms of speed. That is faster; encryption and decryption of data. Because MVECC scheme needs to compute two inverses for each decryption process, and the (Al-Saffar and Said's algorithm) needs to perform the inverse

operation only once for each decryption process , whereas Algorithm I and Algorithm II does not calculate the inverse operation. So the algorithm I and algorithm II is the best based on the execution time of encryption and decryption compared to both (Al-Saffar and Said's algorithm) and MVECC algorithm. Still, (Al-Saffar and Said's algorithm) is more efficient than MVECC algorithm.

Chapter Five

Conclusions

5.1 Conclusions

Elliptic Curve Cryptography provides greater security and more efficient performance than the first generation public key techniques in use now, such as (RSA and Diffie-Hellman). Elliptic curves are believed to provide good security with smaller key sizes, something that is very useful in many applications. Smaller key sizes may result in faster execution times for the schemes, consider the elliptic curve alternative for the computational and bandwidth.

After this, it is clear that the MVECC scheme has another advantage; that it never needs to embed the plaintext as points in the elliptic curve which is the case in ElGamal scheme. This makes it more efficient than the original ElGamal scheme.

In this study the performance of the original algorithm, which rely on elliptical curve, has been improved. Where two variations of the MVECC were proposed, these variations show an improved performance in terms of decryption. That is faster decryption of data. This makes the algorithms more efficient and reduce the total execution time for (encryption and decryption). Because algorithm I and algorithm II does not calculate the inverse operation in decryption process, and the (Al-Saffar and Said's Algorithm) needs to perform the inverse operation only once for each decryption process whereas MVECC scheme need to compute the two inverse for each decryption process. So the algorithm I and algorithm II are the best in terms of the execution time of encryption and decryption compared with both MVECC and Al-Saffar and Said algorithm.

5.2 Suggestions for future work

We considers that there are many areas of research within the topic of Cryptography Based on Elliptic Curve like:

- Applying of the proposed algorithm and the MVECC algorithm on the images file format and calculate the running times of encryption and decryption process.
- Using C# language rather than Java to implement the simulation software of the proposed algorithms and MVECC algorithm and calculate running times in the encryption and decryption process. As the C# language that may be faster in obtaining the results.
- The time it takes to encrypt and decrypt the text file may be improve; that is reduced if data is compressed before the encryption process takes place.
- Implementing the algorithms proposed in this research using hardware description languages and tested on a FPGA device.

References

1. B. P Fitzmann, "Trials of traced traitors." Information hiding, first international work shop, Lecture notes in computer science R. Anderson, Ed. Berlin, Germany: Springer Verlag, Vol. 1, pp= 49-64, 1996.
2. W. Stallings, "Cryptography and Network Security Principles and Practices", 5th Edition, Prentice Hall, 2011.
3. Al-Saffar, N. F. and Md-Said, M. R. "On the Mathematical Complexity and the Time Implementation of Proposed Variants of Elliptic Curves Cryptosystems", International Journal of Cryptology Research, Vol. 4(1), pp.42 – 54, 2013.
4. Al-Saffar and Najlae Falah Hameed, "Proposed Developments of Elliptic Curves Cryptosystem", Master's thesis, University of Babylon, Iraq, 2005.
5. Kadhim, N. J. and Sadiq, A. T., "Enhanced Menezes-Vanestone elliptic curves cryptosystem", Journal of Al-Nahrain University, Vol. 12 (1), pp. 162-165, 2009.
6. C. Paar and J. Pelzl, "Understanding Cryptography", 1st Edition, Springer, 2010.
7. Bruce Schneier, "Applied Cryptography", 2nd Edition, John Wiley & Sons, Inc, 1996.
8. Massey LJ, Omura JK. OMNET, "Method and Apparatus for Maintaining the Privacy of Digital Messages Conveyed by Public Transmission", Filling date: 1982-09-14, United States patent # 4567600, 1986.
9. ElGamal T. A, "Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory, Vol. 31, pp. 469-72, 1985.
10. Koblitz N., " Elliptic curve cryptosystems", Math Computation, Vol. 48, pp. 203-9, 1987.
11. Menezes A, Vanstone S., "Elliptic curve cryptosystems and their implementation", J Cryptol, Vol. 6, pp. 209-224, 1993.
12. Song Y. Yan, "Computational Number Theory and Modern Cryptography", 1st Edition, John Wiley & Sons Singapore Pte, 2013.
13. Pietilainen, H., "Elliptic Curve Cryptography on smart cards", Master Thesis, University of Technology, Helsinki, 2000.

14. Saeki, M.K. ,“Elliptic Curve Cryptosystems”, Master Thesis, McGill University, Montreal, 1997.
15. Song Y. Yan, “Number Theory for Computing”, 2nd Edition, Springer, 2000.