

## SIMULATION OF ECC OVER BINARY FIELD



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

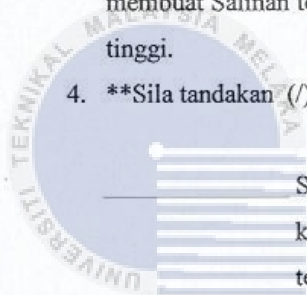
## BORANG PENGESAHAN STATUS TESIS

JUDUL : Simulation of ECC over Binary Field

SESI PENGAJIAN : 2016/2017

Saya NUR-SUHADA BINTI MASURI Mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut :

1. Tesis dan projek adalah hak milik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat Salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat Salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **\*\*Sila tandakan (/)**



SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan)

TIDAK TERHAD

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

*Nur Suhada*

(TANDATANGAN PENULIS)

Alamat tetap: No 22, Jalan 2/3,  
Taman Puteri,  
86000 Kluang, Johor.

Tarikh: 22-08-2017

*Prof Madya Dr Nor Azman Bin Abu*

(TANDATANGAN PENYELIA)

Prof Madya Dr Nor Azman Bin Abu

Nama Penyelia

Tarikh: 22/8/17

CATATAN: \* Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda

\*\* Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

# SIMULATION OF ECC OVER BINARY FIELD

NUR-SUHADA BINTI MASURI




UNIVERSITI TEKNIKAL MALAYSIA MELAKA

This report is submitted in partial fulfilment of the requirements for the  
Bachelor of Computer Science (Computer Security)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY  
UNIVERSITI TEKNIKAL MALAYSIA MELAKA  
2017

## DECLARATION

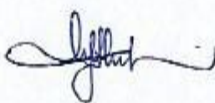
I hereby declare that this project report entitled  
**SIMULATION OF ECC OVER BINARY FIELD**  
is written by me and is my own effort and that no part has been plagiarized without  
citations.

STUDENT :   
(NUR-SUHADA BINTI MASURI) Date : 22/08/2017

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

I hereby declare that I have read this project report and found this project report is  
sufficient in term of the scope and quality for the award of Bachelor of Computer  
Science (Computer Security) With Honours.

SUPERVISOR :   
(PROF MADYA DR NOR AZMAN  
BIN ABU) Date : 22/8/17

## DEDICATION

To my beloved parents for their supports and external love.



## ACKNOWLEDGEMENT

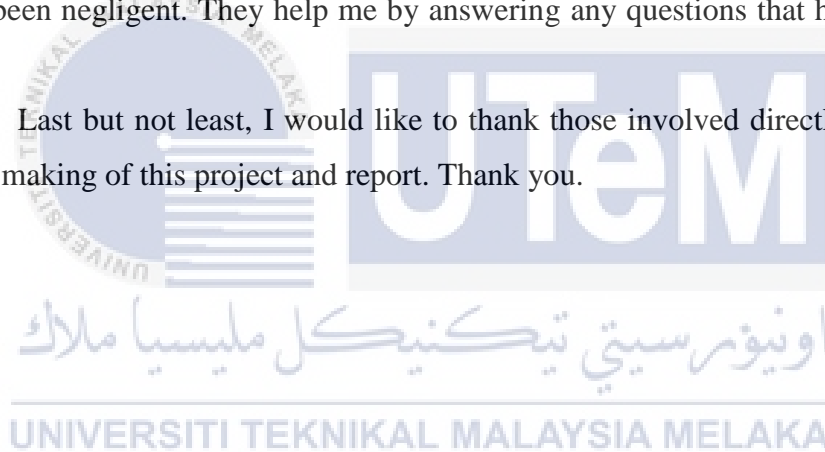
Thank God, give thanks to the divine mercy and grace period, the life energy can also be awarded to me I completed this project successfully.

Firstly, I would like to dedicate this award to my beloved supervisor, Professor Madya Dr Nor Azman bin Abu as a tutoring and mentoring me to complete this project with success.

Furthermore, I would like to express my deepest appreciation to my parents who gave me a facilitator to complete this project. They have given me all the encouragement and moral support to complete on this project.

This gratitude also goes to the many friends who warn against any of what I have been negligent. They help me by answering any questions that had been ask to them.

Last but not least, I would like to thank those involved directly or indirectly in the making of this project and report. Thank you.



## ABSTRACT

Elliptic Curve Cryptography (ECC) is a method of public key cryptography, which develop by Diffie Hellman. The Diffie-Hellman key exchange protocol and the Digital Signature Algorithm (DSA) based on it, is an asymmetric cryptographic system used today. Elliptic curves over the real numbers on the axis  $xy$  plot. Traditionally, the ECC is one of key areas textbooks fifty years modulo primes. However, the most practical binary field ECC taken modulo a polynomial can be reduced. Nowadays, it is mainly used in resource-constrained environments, such as wireless networks and ad-hoc mobile networks. By 2000, there is a tendency that the conventional public key cryptography system, especially the RSA-based system, gradually replaced by ECC system. However, today other cryptographic NTRU contender makes its way into the mainstream market. The problem statement of this project are the numbers that will be calculated usually a limited numbers, the calculation on ECC over binary field before this is still in manually managed by user and user can compute only a small numbers to calculate ECC over binary field. The objectives of this project are to generate random numbers during calculation, to build and develop a system to calculate ECC over binary field for user and to enable the user to calculate or compute large numbers on ECC over binary field. The methodology used for this project is V-Shaped model which consists the process of requirements or analysis, design, implementation, testing and maintenance. The significant contribution of this project is to simplify and improve the efficiency of the simulation ECC on binary field, minimize manual data entry and ensure data accuracy and security during calculation process.

## ABSTRAK

Kriptografi Kurva Elliptic (ECC) adalah kaedah kriptografi utama awam, yang dibangunkan oleh Diffie Hellman. Protokol pertukaran utama Diffie-Hellman dan Digital Signature Algorithm (DSA) adalah sistem kriptografi asimetri yang digunakan hari ini. Keluk eliptik ke atas bilangan sebenar pada plot paksi  $xy$ . Secara tradisinya, ECC adalah salah satu bidang utama buku teks lima puluh tahun modulo primes. Walau bagaimanapun, bidang binari yang paling praktikal ECC yang diambil modulo polinomial boleh dikurangkan. Pada masa kini, ia digunakan terutamanya dalam persekitaran terkurung sumber, seperti rangkaian tanpa wayar dan rangkaian mudah alih ad hoc. Menjelang tahun 2000, terdapat kecenderungan sistem kriptografi utama awam konvensional, terutama sistem berasaskan RSA, secara beransur-ansur digantikan oleh sistem ECC. Walau bagaimanapun, hari ini pesaing NTRU kriptografi lain membuat jalan ke pasaran arus perdana. Pernyataan masalah projek ini adalah bilangan yang akan dikira biasanya nombor terhad, pengiraan pada ECC berbanding medan binari sebelum ini masih diurus secara manual oleh pengguna dan pengguna dapat mengira hanya sejumlah kecil untuk menghitung ECC atas medan binari. Objektif projek ini adalah untuk menghasilkan nombor rawak semasa pengiraan, untuk membina dan membangun sistem untuk mengira ECC ke atas bidang perdua untuk pengguna dan membolehkan pengguna untuk mengira atau mengira bilangan besar pada ECC berbanding medan binari. Metodologi yang digunakan untuk projek ini ialah model bentuk V yang merangkumi proses keperluan atau analisis, reka bentuk, pelaksanaan, pengujian dan penyelenggaraan. Sumbangan besar projek ini adalah untuk memudahkan dan meningkatkan kecekapan simulasi ECC pada bidang binari, meminimumkan kemasukan data manual dan memastikan ketepatan dan keselamatan data semasa proses pengiraan.



## TABLE OF CONTENTS

BORANG PENGESAHAN STATUS TESIS .....	ii
DECLARATION .....	<b>Error! Bookmark not defined.</b>
DEDICATION .....	v
ACKNOWLEDGEMENT .....	vi
ABSTRACT .....	vii
ABSTRAK .....	viii
TABLE OF CONTENTS .....	ix
LIST OF TABLES .....	xiii
LIST OF FIGURES .....	xiv
LIST OF ABBREVIATION .....	xv
CHAPTER 1 .....	1
INTRODUCTION .....	1
1.1 Project Background .....	1
1.2 Problem Statement .....	2
1.3 Project Question .....	2
1.4 Project Objectives .....	3
1.5 Project Scope .....	3
1.5.1 Programming language .....	4
1.5.2 Algorithm .....	4
1.5.3 User .....	4
1.6 Project Contribution .....	4
1.7 Expected Output .....	5
1.8 Thesis Organization .....	5
1.9 Conclusion .....	6
CHAPTER 2 .....	7
LITERATURE REVIEW .....	7
2.1 Introduction .....	7
2.2 Related work .....	8

2.2.1 Elliptic Curve Cryptography (ECC) for Security in Wireless Sensor Network.....	8
2.2.1.1 Architecture and Constraints of Wireless Sensor Networks (WSN).....	9
2.2.1.2 Security Mechanism using ECC.....	10
2.2.1.3 Security Requirements in Wireless Sensor Network (WSN) .....	11
2.3 Critical Review of Current Problem and Justification.....	12
2.3.1 Implementation of Elliptic Curve Cryptography (ECC) on Smart Card.....	12
2.3.1.1 General of Development .....	13
2.3.1.2 Architecture of the Program .....	14
2.3.1.3 Technical Improvements .....	15
2.4 Term Used.....	15
2.5 History of Elliptic Curve Cryptography .....	16
2.6 ECC Modulo on Irreducible Polynomial .....	18
2.7 Software and Hardware .....	23
2.8 Conclusion .....	23
CHAPTER 3 .....	24
METHODOLOGY.....	24
3.1 Introduction.....	24
3.2 Methodology.....	25
3.3 Project Milestone .....	28
3.4 Finite Fields .....	29
3.4.1 Finite Field $F_p$ .....	30
3.4.2 Finite Field $F_2^m$ .....	31
3.5 Recommended Finite Fields .....	35
3.6 Conclusion .....	35
CHAPTER 4 .....	36
DESIGN .....	36
4.1 Introduction.....	36
4.2 System Architecture.....	37

4.2.1 Interface.....	38
4.2.2 Addition Interface .....	39
4.2.3 Multiplication Interface.....	40
4.2.4 Inversion Interface .....	40
4.2.5 Point Addition Interface.....	41
4.2.6 Point Doubling Interface .....	41
4.2.7 Point Multiplication Interface .....	42
4.3 Logical Design.....	43
4.4 Possible Scenario .....	44
4.5 Conclusion .....	46
CHAPTER 5 .....	47
IMPLEMENTATION .....	47
5.1 Introduction.....	47
5.2 Software Development Environment Setup .....	48
5.3 Software Configuration Management.....	49
5.3.1 MATLAB Installation.....	49
5.4 Implementation Status .....	52
5.5 Code of the System.....	54
5.6 Conclusion.....	57
CHAPTER 6 .....	58
TESTING .....	58
6.1 Introduction.....	58
6.2 Test Plan .....	59
6.2.1 Test Environment.....	59
6.3 Test Result and Analysis.....	59
6.3.1 Addition .....	60
6.3.2 Multiplication.....	60
6.3.3 Inversion.....	61
6.3.4 Point Addition.....	61
6.3.5 Point Doubling .....	62
6.3.6 Point Multiplication .....	63
6.4 Conclusion .....	63

CHAPTER 7 .....	64
CONCLUSION .....	64
7.1 Introduction.....	64
7.2 Project Summarization.....	64
7.3 Project Contribution.....	66
7.4 Project Limitation .....	66
7.5 Future Works .....	66
7.6 Conclusion .....	67
REFERENCES.....	68
BIBLIOGRAPHY .....	70



## LIST OF TABLES

<b>Table 1.1 : Summary of Problem Statement.....</b>	<b>2</b>
<b>Table 1.2 : Summary of Problem Question.....</b>	<b>3</b>
<b>Table 1.3 : Summary of Project Objectives.....</b>	<b>3</b>
<b>Table 1.4 : Summary of Project Contribution.....</b>	<b>5</b>
<b>Table 2.1 : Layer Structure.....</b>	<b>15</b>
<b>Table 3.1 : Gantt Chart.....</b>	<b>29</b>
<b>Table 3.2 : Milestone Activities.....</b>	<b>30</b>
<b>Table 3.3 : Igcd.....</b>	<b>35</b>
<b>Table 5.1 : Implementation Status.....</b>	<b>54</b>
<b>Table 6.1 : Test Environment Specification.....</b>	<b>61</b>



## LIST OF FIGURES

Figure 2.1 : Architecture of Wireless Sensor Network.....	9
Figure 2.2 : Various Security Schemes in WSN.....	11
Figure 2.3 : Simplified UML Diagram of the ECC Engine.....	14
Figure 3.1 : V-Shaped Model.....	26
Figure 4.1 : ECC Operational Pyramid.....	39
Figure 4.2 : Arithmetic Dependency.....	40
Figure 4.3 : Main Menu Interface.....	41
Figure 4.4 : Addition Interface.....	41
Figure 4.5 : Multiplication Interface.....	42
Figure 4.6 : Inversion Interface.....	42
Figure 4.7 : Point Addition Interface.....	43
Figure 4.8 : Point Doubling Interface.....	44
Figure 4.9 : Pont Multiplication Interface.....	44
Figure 4.10 : Flowchart.....	45
Figure 5.1 : Software Development Environment Setup Architecture.....	50
Figure 5.2 : Deployment Diagram.....	51
Figure 5.3 : MATLAB File in the Documents Folder.....	51
Figure 5.4 : Matlab803 File.....	52
Figure 5.5 : Warning Message MATLAB.....	52
Figure 5.6 : Setup.....	52
Figure 5.7 : Select Component.....	53
Figure 5.8 : Matlab Icon.....	53
Figure 5.9 : Matlab Main Page.....	54
Figure 6.1 : Result of the Operation of Addition.....	62
Figure 6.2 : Result of the Operation of Multiplication.....	62
Figure 6.3 : Result of the Operation of Inversion.....	63
Figure 6.4 : Result of the Point Addition.....	64
Figure 6.5 : Result of the Point Doubling.....	64
Figure 6.6 : Result of the Point Multiplication.....	65

## LIST OF ABBREVIATION

ECC – Elliptic Curve Cryptography

RSA – Rivest-Shamir-Adleman

DH – Diffie-Hellman

ECDSA – Elliptic Curve Digital Signature Algorithm

DLP – Discrete Logarithm Problem

PDA's – Personal Digital Assistants

NTRU – Number Theorists are Us

WSN – Wireless Sensor Networks

ECDLP – Elliptic Curve Discrete Logarithm Problem

SDLC – System Development Life Cycle

DSA - Digital Signature Algorithm



## CHAPTER 1

### INTRODUCTION

#### 1.1 Project Background

Elliptic Curve Cryptography (ECC) is a method of public key cryptography, which develop by the Diffie Hellman. The Diffie-Hellman key exchange protocol and the Digital Signature Algorithm (DSA) based on it, is asymmetric cryptographic. In 1976, it was found by Whitfield Diffie and Martin Hellman and the problem known as the Discrete Logarithm Problem (DLP) as the asymmetry operations. DLP finding concerns the logarithm of a number in the arithmetic of finite fields. Prime field is the field which is the sets are prime. In other words, they have a prime number of members. The "Diffie-Hellman Method for Key Agreement" enable the two groups to create and share a secret key. However, ECC can provide the highest security per bit than other public-key cryptosystems traditional such as RSA and DH. ECC has a throughput rate is usually lower and the most complex calculations. ECC interesting feature makes it very popular for resource-constrained applications such as smart cards, credit cards, pagers, Personal Digital Assistants (PDAs) and mobile phones

Elliptic curves over the real numbers plots on the  $xy$  axis. Traditionally, ECC over a prime field follows the 50 years old textbook modulo a prime number. However, today the most practical ECC over binary field takes modulo an irreducible polynomial. In 1985, Neal Koblitz and Victor Miller independently proposed using elliptic curves to design a public key cryptography system. In the late 1990's, ECC has been standardized by several organizations and it began receiving commercial acceptance. Nowadays, it is mainly used in the resource-constrained environments, such as ad-hoc wireless networks and mobile networks. By 2000, there is a tendency that the conventional public key cryptography system, especially the RSA-based



system, gradually replaced by ECC systems. However, in 2016, other contender cryptographic system NTRU makes its way into the mainstream market.

## 1.2 Problem Statement

The problem statement to develop the system are because the numbers that will be calculated usually limited to small numbers. For teaching and learning purpose, lecturers and students usually need to calculate the operation using large numbers. Other than that, the academic computation on ECC over binary field before this is still calculated manually and managed by user. In industrial, they are doing the computation for ECC over binary field using computerized system, but in university, we still compute by manual calculation. The summary of problem statement is described as shown in Table 1.1.

**Table 1.1 : Summary of Problem Statement**

PS	Problem Statement
PS1	The numbers that will be calculated usually limited to small numbers.
PS2	The academic computation on ECC over binary field before this is still calculated manually and managed by user.

## 1.3 Project Question

Smaller chip size, less power consumption and increase in speed is required for the implementation of elliptic curve in Cryptography. Elliptic Curves contribute security to classical system but uses fewer bits. The summary of problem question is shown in Table 1.2.

**Table 1.2 : Summary of Problem Question**

PQ	Problem Question
PQ1	How Elliptic Curve Cryptography will help in securing the data.

## 1.4 Project Objectives

The significance of this project is to make sure that this simulation of ECC over binary field is successful to test and able to validate the accuracy and performance. The objectives of this project is to generate random numbers during calculation. This system will be generating random numbers instead of enter the numbers manually. Other than that, the objectives of this project is to build and develop a system to calculate cryptosystem of ECC over binary field for user. The computation of ECC over binary field in academic before this is still in manual way, so instead of calculate manually, this system will be develop to facilitate the user. Lastly, the objective of this project is to enable the user to calculate or compute large numbers on ECC over binary field. The computation of ECC over binary field not using only small numbers. The system is needed to compute large numbers that is impossible to be calculated in manual ways. The summary of project objectives is shown in Table 1.3.

**Table 1.3 : Summary of Project Objectives**

<b>PS</b>	<b>PQ</b>	<b>PO</b>	<b>Project Objectives</b>
<b>PS1</b>		<b>PO1</b>	To generate random numbers during calculation.
<b>PS2</b>	<b>PQ1</b>	<b>PO2</b>	To build and develop a system to calculate cryptosystem of ECC over binary field for user.
		<b>PO3</b>	To enable the user to calculate or compute large numbers on ECC over binary field.

## 1.5 Project Scope

Project scope is the part of project planning that involves to determine and document a list of specific project goals, programming language used, tasks, function, algorithm, software or tools and who are the user. Project scope have the ability to achieve the objective of the project by focussing on the specific scope.

### 1.5.1 Programming language

This project use language of MATLAB programming as it is suitable for this project. MATLAB is one of the programming language that are very easy and most productive to analyze data, developing the algorithm and creating models. This project covered the underlying fields of binary field only for cryptosystem of ECC.

### 1.5.2 Algorithm

This project use algorithm of Elliptic Curve Cryptography over binary field algorithm by using MATLAB 2014a version.

### 1.5.3 User

This system have target users who are students and lecturer which able to used for learning and teaching process and the organization that want to verify that the data stored is secured and for the fast performance.

### 1.6 Project Contribution

This project will contribute to the user to teach and learn how the operation which are addition, multiplication and inversion and for the representation point are point addition, point doubling and point multiplication are work by this simulation of ECC over binary field. The summary of projection contribution is shown in Table 1.4.

**Table 1.4 : Summary of Project Contribution**

PS	PQ	PO	PC	Project Contribution
PS1	PQ1	PO1	PC3	Proposed a method of teaching and learning using simulation.
PS2		PO2		
		PO3		

## 1.7 Expected Output

This system will have the interface that :

- i. Allow the user to choose which type of operation they want to calculate or use.
- ii. Allow user to enter an irreducible binary polynomials and degree  $m$ .
- iii. Display the results in every operation which are addition, multiplication and inversion. For representation point are point addition, point doubling and point multiplication.

## 1.8 Thesis Organization

This topic is about the summary of each chapter that will be introduced in this report later.

### Chapter 1 : Introduction

For the first chapter which is introduction, will discuss about eight subtopic which are problem statement, project question, project objectives, project scope, project significance, expected output, thesis organization and lastly is conclusion.

### Chapter 2 : Literature review

For the second chapter which is literature review, will discuss about the previous work or related work that related with this project, critical review of current problem and recommended solution.

### Chapter 3 : Project methodology

For the third chapter which is project methodology, will discuss about the project methodology used in the process of developing the system and project milestones.

#### **Chapter 4 : Analysis and design**

For the fourth chapter which is analysis and design, will discuss about the network system architecture, logical and physical design and possible scenario.

#### **Chapter 5 : Implementation**

For the fifth chapter which is implementation, will discuss about how the environment is setup.

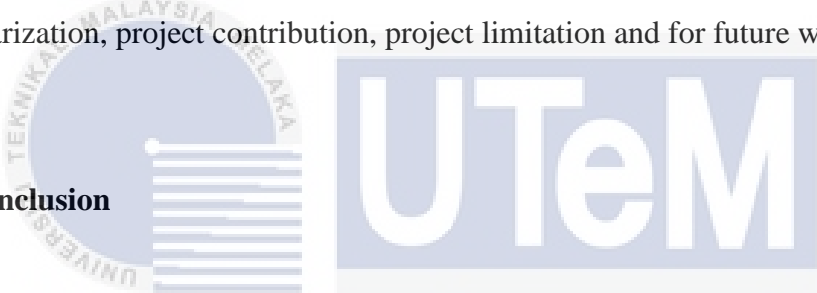
#### **Chapter 6 : Testing**

For the sixth chapter which is testing, will discuss about the final result and analysis.

#### **Chapter 7 : Project conclusion**

For the last chapter which is project conclusion, will discuss about the project summarization, project contribution, project limitation and for future work.

### **1.9 Conclusion**



As a conclusion, this system has a function to calculate every single operation which are addition, multiplication and inversion which to ease the user who are still compute in manually. This system is developed to hope that it is able to serve as learning and teaching tool. After this, the next step is to develop the system. To develop this system, programming language used is Matlab R2014a. Next, the chapter that will be discussed is literature review and project methodology which based on the articles and journals found. To develop this system, we need to discuss about the previous resarch and the methodology used.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Introduction

For chapter two, the literature review main topic that are being discussed. For this main topic which is literature review, there are three sub-topics will reflect particularly on the issues and topics related to Elliptic Curve Cryptography (ECC). For the first part, it is about the previous study that was done by the other researchers. The last part is focused on critical review of the current problem and justification. Lastly, for the last part, it is about the software and hardware used for this project to develop the system. These literature review could be found from any reading materials such as reference books and websites which based on some journals, thesis and findings. Therefore, when reviewing the literature, it can be an opportunity to read and study in depth about what is public key algorithms all about.

This chapter is requiring the process of searching, reading, collecting, analyzing and making a conclusion based on the findings. Observations and research materials also can be used as a reference in developing the system in this project. The system that already existed also need to elaborate to provide a clearer image what is the project is about, the benefits and its limitation and also how the programs or the system is work.

## **2.2 Related work**

There are many existing systems that related to the Elliptic Curve Cryptography (ECC). Some of the systems have been already in the market while some of the systems are still in the form of research. One of the examples of the systems is Elliptic Curve Cryptography (ECC) for Security in wireless Sensor Network

### **2.2.1 Elliptic Curve Cryptography (ECC) for Security in Wireless Sensor Network**

Wireless Sensor Networks (WSNs) thrived in their interest and relevance to the research community and citizen. Safety or security is important for a variety of sensors network applications. There are many security weaknesses in WSNs, which cause various types of attacks. Wireless Sensor Networks (WSN) is a network that represent using small and low power sensor devices. There are two types of communication occurs in sensor networks which are the first is between end node and the second one is between the node and the base end station (BS). The important issues in Wireless Sensor Network (WSN) are security functions and security critical applications.

The main recommended security in Wireless Sensor Network (WSN) is not the message encryption but it is prevented the changes of content message or hide the sender. It is the most important joint confirmation to protect from the sender who is pretending. Due to the limited energy, many studies have been conducted to maximize the lifespan of the networks.

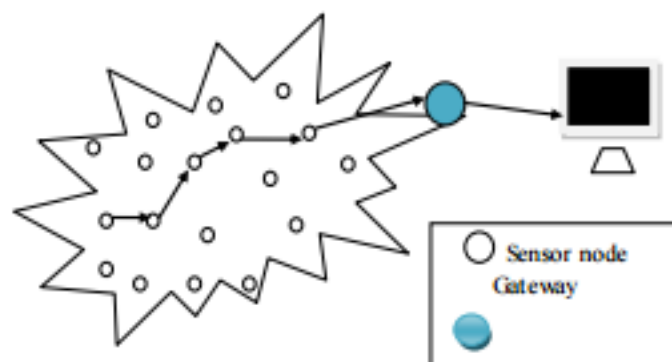
The implementation of symmetric key algorithm is suitable for Wireless Sensor Network (WSN) resource constrained environment. However, symmetry algorithms provide confidentiality only. Confidentiality and the other security issues is provided by public key cryptosystems. Researcher interested in ECC as it provides smaller key size. It offers practical possible implementation in resource-constrained devices. Previous work shows that a public key algorithm is a suitable choice to use in wireless sensor networks and that the advantages of the ECC has smaller key sizes

and certificate will be less important in increasing the energy conservation. By providing the authentication and key management are how the ECC is being used.

ECC and related work on wireless communication is based on elliptic curve cryptography techniques. At this time, RSA algorithm demands the key length that is not less than 1024 bit to long-term security. Meanwhile, ECC with only 160 bit modulus provide the same level of security as RSA with a 1024 bit modulus. Therefore, using ECC in wireless communication system is highly recommended. Distribution of the key and storage are the main problems, which are common in secret-key settings, it solved with ECC cryptography concept. (Mishra, Asha Rani, 2012)

### 2.2.1.1 Architecture and Constraints of Wireless Sensor Networks (WSN)

Sensor network consists of ten to thousands of sensor nodes that are placed over a wide area. All the nodes communicate with each other either directly or through another node. Using one or more nodes among those nodes considered as sink. All other nodes in the network need to send the data to the sink. Architecture of wireless sensor networks, including both hardware platforms and operating systems that are designed specifically to meet the needs of wireless sensor networks. There are many constraints of Wireless Sensor Network which are energy constraints, memory limitations, the communication is unreliable and need higher latency in communication. (Mishra, Asha Rani, 2012)



**Figure 2.1 : Architecture of Wireless Sensor Network**



### 2.2.1.2 Security Mechanism using ECC

Due to the exposure of wireless sensor networks, secure communication between nodes is compulsory. The Elliptic Curve Cryptography (ECC) is based on the algebraic concepts related to elliptic curves over finite fields  $F_p$  or  $F_2^m$ . Elliptical Curve encryption and decryption system needs appointing  $G$  and elliptic group  $E_q(a, b)$  as a parameters. (Mishra, Asha Rani, 2012)

#### Encryption using ECC

To encrypt and send a message  $Pm$  to user B. User A chooses a random positive integer  $k$  and generate the cipher text  $Cm$  as given by the equation which consist the pair of points.

$$Cm = [k * G, Pm + k * P_B]$$

User A has used user B's public key,  $P_B$

#### Decryption using ECC

To decrypt the cipher text, user B multiples the first point in the pair by B's private key  $n_B$  and subtracts the result from the second point as shown by the equation.

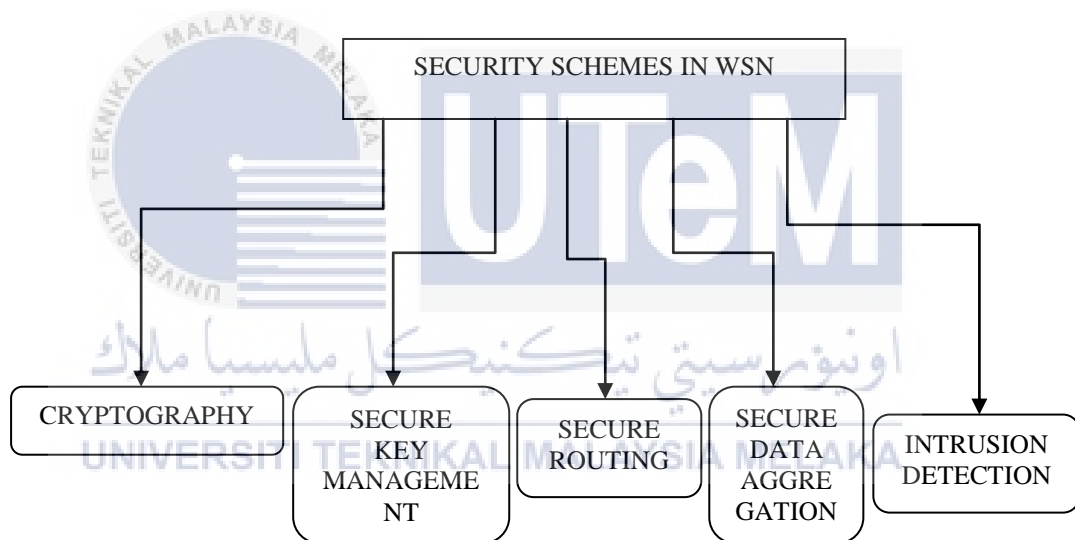
$$Pm + k * P_B - n_B (k * G) = Pm + k(n_B * G) - n_B (k * G) = Pm$$

Steps of the exchange of key between users A and B :

- 1<sup>st</sup> step – User A choose an integer  $n_A < n$  as user A's private key.
- 2<sup>nd</sup> step – User A generates a public key  $P_A = n_A * G$  which is point in  $E_q(a, b)$ .
- 3<sup>rd</sup> step – User B choose an integer  $n_B < n$  as user B's private key.
- 4<sup>th</sup> step – User B generates a public key  $P_B = n_B * G$  which is point in  $E_q(a, b)$ .
- 5<sup>th</sup> step – User A and B exchanges their public key. User A generates the secret key  $K = n_A * P_B$  and user B generates the secret key  $K = n_B * P_A$

### 2.2.1.3 Security Requirements in Wireless Sensor Network (WSN)

Many of sensor network application have a critical issue of security issue. There are numerous number of vulnerabilities in WSNs such as intrusion, interception, modification and fabrication which affects to a number of threats to WSN protocols. The threat can be registered from different perspectives. The list of threat on how attacks are achieved, in which layer of the communication stack that they are aware and lastly during the attack, are there any malicious node that becomes the member of the network have been listed by the previous research. Multiple security issues in Wireless Sensor Network (WSN) can widely classified as Cryptography, key management, secure routing, data aggregation and intrusion detection.



**Figure 2.2 : Various Security Schemes in WSN**

To ensure WSNs is secured, there are security objectives that give security services, such as a Confidentiality which all the confidential information can not be disclosed, the authenticity is required in the network of sensors for each sensor node and the base station to make sure that the data received was sent by a trusted person or party. Authentication is required during grouping the sensor nodes in WSN. The integrity of information should always be sure that the information will not be changed in such an unexpected way.

Based on symmetric and asymmetric algorithm, they are a large number of cryptographic solutions that have been proposed until now. As for symmetric algorithm, it provides confidentiality while executing the power, space and memory requirements of WSN. Despite that, they failed to provide authenticity or in other words is legitimacy and right key exchange mechanism to achieve through public key cryptography. (Mishra, Asha Rani, 2012)

## **2.3 Critical Review of Current Problem and Justification**

This subtopic consists the example of technique, parameter or attributes and software and hardware used in previous research that related with this project.

### **2.3.1 Implementation of Elliptic Curve Cryptography (ECC) on Smart Card.**

Smart cards is a small device with the shape and size of a credit card. The microchip that integrated on them is the reason why they called it 'smart'. A CPU, non-volatile memory, and I / O peripherals was embedded on this chip cards. These smart card were labelled as a standardized secure portable microcomputers. Since ECC operate with significantly shorter key compared to RSA as both providing the same security, it can be a great alternative to RSA in low-resource platforms, such as smart cards. This section will describes the smart card based on the implementation of ECC.

With the implementation of ECC, to generate key pairs, it only take a short time that even with a very limited to compute power of a smart cards can generate a key pair, given a good random number generator is available. This means that the personal card can be adjusted for applications where non-repudiation is important. (Berta, Istvan Zsolt, & Mann, Zoltan Ádam2003)

### 2.3.1.1 General of Development

The main purpose of this implementation is to study ECC and conduct certain measurements and calculations and create a smart card based on prototype ECC implementation. Usually, the performance required for commercial use cannot have in the software-based solution. The research focus on to prove that complex algorithms such as ECC can be implemented on a weak smart card nowadays. Another purpose of this study was to provide a solution that can be used as a basis for designing hardware acceleration. It is definitely not the real goal to display an ECC implementation fast enough for commercial use, since the implementation of genuine software cannot keep up with the hardware-acceleration. Development priority is to implement efficient algorithms with polynomial complexity in low-resource smart cards.

This program is able to run on a PC and Java Card. This means that the same source code and the same Java classes run on both platforms. This means the researchers are aiming to not optimize the program to any platform including a Java VM, but to implement a solution that capable of running on both platforms. Due to limited amount of card memory, speed need to be trade in order to get memory. (Berta, IstvanZsolt, & Mann, Zoltan Ádam 2003)

### 2.3.1.2 Architecture of the Program

By taking the benefits from object oriented in Java, this program was designed in a modular form. The researcher able to do an experiment with variety of finite field and variety of representations in the memory card. The structure of the ECC engine make it easier to change to another Galois field.

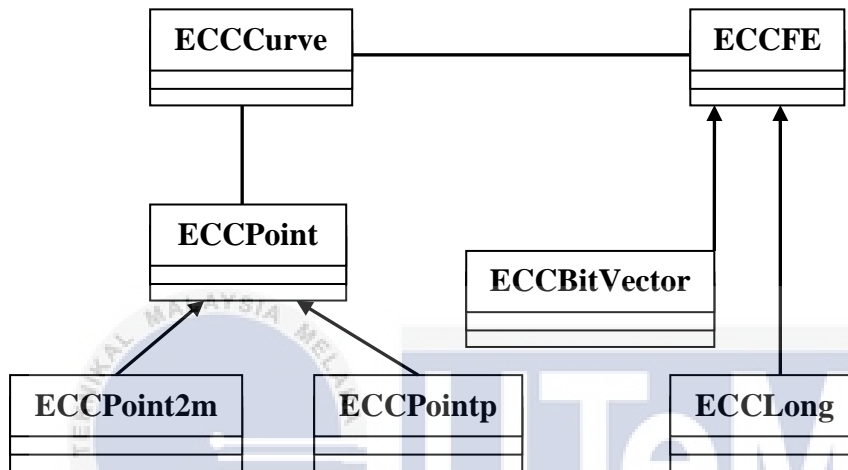


Figure 2.3 : Simplified UML diagram of the ECC engine

The field elements arithmetic is independent, so it can easily being exchanged by another arithmetic. The top position of the field arithmetic layer is the point-arithmetic layer which implements operations between points of the curve. ECDLP-based protocols is implemented by the ECC engines which is the top-level layer. It is completely independent of the selected field.

Table 2.1 : Layer Structure

<b>ECC engine</b>
<b>Curve arithmetics</b>
<b>Field arithmetics</b>
<b>Java virtual machine</b>
<b>Smart card hardware</b>

The application uses the engine that can only access the upper layer, that grants access only to public data which are the curve, the public point and the public key users and deny access to private or personal data. In fact, this layer provides the services such as control, encryption, decryption and signature. The responsibility of this layer is to make sure that the engine works safely and stable and with the right parameters.

In addition, the application of a particular layer can be placed by developers who want to use the engine. The responsibility of this layer is to perform application-specific functions depending on the operations of the ECC engine : (Berta, Istvan Zsolt, & Mann, Zoltan Ádam 2003)

- i. Define who may access the ECC engine
- ii. Conduct the authentication of the above principles
- iii. Assign operations to the ECC engine
- iv. Organize an application-specific data

### **2.3.1.3 Technical Improvements**

To make the ECC algorithm work on a smart card platform is the main aim in their implementation. The example of side channel attacks which is the analysis on key size dependent execution times or analysis on power consumptions or electromagnetic field is not protected through their implementation. In fact, all of these need optimization for a specific smart-card. (Berta, Istvan Zsolt, & Mann, Zoltan Ádam 2003)

## **2.4 Term Used**

There are some terms used in this project that provide function as a keywords in doing this researches. Below are the some keyword that have been define :

i) Cryptography

Cryptography is the study of mathematical techniques related to aspects on information security such as confidentiality, data integrity, entity authentication and data origin authentication. (Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, 1996) It means the information which includes data or files that need to send to the recipient will be secured by using this technique.

ii) Symmetric

It is the process of encrypting plaintext into ciphertext. To provide security and privacy, encryption algorithm is use by cryptography. In fact, there are three categoies of encryption algorithms which are symmetric, assmetric and digest algorithms (Skalli, Bulus, Buyuksaracoglu, 2004)

iii) Public key cryptography

Public key cryptography is a “one-way” process or function mathematical which the inverse cannot feasibly be computed (Minal Wankhede Barsagade, Dr. Suchitra Meshram, 2014)

## 2.5 History of Elliptic Curve Cryptography

Elliptic curves and their properties have been studied in mathematics as a pure mathematical concept of old since the second or third century A.C. but its use in cryptography is recently. In nineteenth century, the name was given as “elliptic” although it has been extensively studied by many mathematicians. Before 1984, the uses of elliptic curve cryptography is not being recognized. The first application in the field of cryptography was established in the method of integer factorization by Lenstra.

In 1985, Victor Miller and Neal Koblitz recommend a completely different the use of elliptic curve cryptography. Elliptic Curves Cryptography (ECC) is a public key cryptography. From mathematical group, a particular type of equation is created to compute ECC. The equations based on elliptic curve is characterized as a very valuable for cryptographic purposes.

Main reason ECC is preferred because it is the fact that no sub exponential algorithm known to solve the discrete algorithms problem. This means that the parameter used can be smaller in the ECC with the same level of security. Elliptic curve as the base for a new class relative of public key scheme. It is assumed that the Elliptic Curve will be replacing many of the existing schemes in the future. (Minal Wankhede Barsagade, Dr. Suchitra Meshram, 2014)

A technology specially called Elliptic Curve Cryptography (ECC), has been the preferred cryptography for mobile computing and communication devices due to its benefits and efficiency. Elliptical curve creates a small and highly efficient computing, which makes it ideal for smaller, less powerful devices today used by the majority of individuals to access network services. Its efficiency allows wireless devices to achieve secure end-to-end connections.

Koblitz and Miller freely recommended Elliptic Curve Cryptosystem (ECC) method a crypto-based algorithm using the Discrete Logarithm Problem (DLP) above the point on the elliptical curve. With their recommendations, the ECC can be used to provide digital signatures and encryption schemes. Over the last decade, the ECC and later ECDLP (Elliptic Curve Logarithm Disorder Problems) have gained considerable attention from mathematicians around the world and there has been no significant discovery in determining weaknesses in algorithms and has now occurred a Mathematical Attack.

The elliptical curved cryptosystem has been accepted today as the most feasible public key technology for high security applications. They are also best suited for confined surround devices such as smartcards and personal wireless devices are usually used. Over the years to come, it will continue to be a great requirement for designing and implementing the ECC cryptosystem supplied from



software, protocols and hardware solutions to secure advanced technologies such as smartcards, cell phones, browsers, servers, RFID Tag Frequency Identification and environmental sensors.

There are three basic point operations of Elliptic Curve which are listed below :

- i. Point addition :  $P(x,y) + Q(x,y)$
- ii. Point doubling :  $2 * P(x,y)$
- iii. Point (scalar) multiplication :  $k * P(x,y)$ , where  $k \in [1, n-1]$  and  $n$  is the order of the EC base point.  $k * P(x,y) = P + P + \dots + P$ . Point multiplication is the fundamental and the most time consuming operation in ECC. It also require the operation of point addition and point doubling. Various algorithms available which are the field type and coordinate representation dependent.

## 2.6 ECC Modulo on Irreducible Polynomial

The Weierstrass equations can be simplified by performing the following change of variables:

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

Cryptographic schemes need fast and accurate arithmetic operations. In the cryptographic schemes, elliptic curves over two finite fields are mostly used.

There are 3 types of ECC, namely, P-curves(P-192, P-224, P-256, P-384, P-512), K-curves(K-163, K-233, K-283, K-409, K-571) and Random B-curves(B-163, B-233, B-283, B-409, B-571).

Prime field  $F_p$ , where  $p$  is a prime.  $P$  is of 256-bit, 384-bit and 512-bits.

Binary field  $F_{2^m}$ , where  $m$  is a positive integer starting from 160, 256, 384 and 512.

An elliptic curve  $E$  over the finite field  $F_{2^m}$  is given through the following equation.

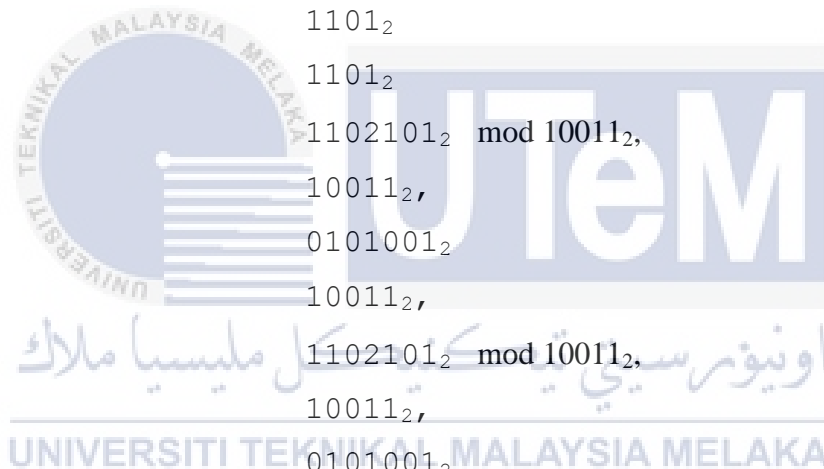
$$y^2 + xy = x^3 + ax + b$$

where the parameters  $a, b \in F_{2^m}$  are m-bit numbers and the points  $(x, y)$  are also m-bit numbers. An identity is a point at infinity  $(+\infty, +\infty)$ .

Let us take a finite field  $F_{2^m}$  over irreducible polynomial  $f(x) = x^4 + x + 1 = 10011_2$ . The element  $g = x = (0010)$  is a generator for the field. The powers of  $g$  are the elements of  $F_{2^m}$ :

$$\begin{aligned} g^0 &= (0001), g^1 = (0010), g^2 = (0100), g^3 = (1000), \\ g^4 &= (0011), g^5 = (0110), g^6 = (1100), g^7 = (1011), \\ g^8 &= (0101), g^9 = (1010), g^{10} = (0111), g^{11} = (1110), \\ g^{12} &= (1111), g^{13} = (1101), g^{14} = (1001), g^{15} = (0001). \end{aligned}$$

Let us take an example  $g^{13} \cdot g^{14} = 1101_2 \cdot 1001_2 \bmod 10011_2$ ,



$$\begin{array}{r} 1101_2 \\ 1101_2 \\ \hline 1102101_2 \bmod 10011_2, \\ 10011_2, \\ 0101001_2 \\ \hline 10011_2, \\ 1102101_2 \bmod 10011_2, \\ 10011_2, \\ \hline 0101001_2 \\ 10011_2, \\ \hline 001111_2 = g^{12} = (1111). \\ g^{13} \cdot g^{14} = g^{27-15} = g^{12}. \end{array}$$

The order here is 15.

The points on  $E: y^2 + xy = x^3 + ax^2 + b$  over irreducible polynomial  $f(x) = x^4 + x + 1$  are

$(1, g^{13}), (g^3, g^{13}), (g^5, g^{11}), (g^6, g^{14}), (g^9, g^{13}), (g^{10}, g^8), (g^{12}, g^{12}), (1, g^6), (g^3, g^8), (g^5, g^3), (g^6, g^8), (g^9, g^{10}), (g^{10}, g), (g^{12}, 0), (0, 1)$  and a point at infinity  $(+\infty, +\infty)$ .

### Point Addition

Let  $P=(x_P, y_P), Q=(x_Q, y_Q)$  on the curve Then  $P+Q=R$  can be computed:

First, compute the slope of the tangent line

$$\lambda = \frac{y_P + y_Q}{x_P + x_Q}$$

Second,  $x_R = \lambda^2 + \lambda + x_P + x_Q + a$  and  $y_R = \lambda(x_P + x_R) + x_R + y_P$

### Point Doubling

Let  $P=(x_P, y_P)$ , compute the slope of the tangent line

$$\lambda = x_P + \frac{y_P}{x_P}$$

Second,  $x_R = \lambda^2 + \lambda + a$  and  $y_R = x_P^2 + \lambda x_R + x_R$

### Point Multiplication

The set of points on  $E(F_{2^m})$  forms an Abelian group under this addition rule. Notice that the addition rule can always be computed efficiently using simple field arithmetic. As before, scalar multiplication is the process of adding  $P$  to itself  $k$  times. The result of this scalar multiplication is denoted  $kP$  and can be computed efficiently using the addition rule together with the double-and-add algorithm or one of its variants.

Let us take an example:

Let the parameters  $a=g^4$  and  $b=1$ ,  $E: y^2 + xy = x^3 + ax^2 + b$  over an irreducible polynomial  $f(x) = x^4 + x + 1$ ,

Let  $P=(g^5, g^3) = (0110, 1000)$ ,  $Q=(g^9, g^{13}) = (1010, 1101)$  on the curve, Then  $R(x_R, y_R)=P+Q$  can be computed:

First, compute the slope of the tangent line

$$\lambda = \frac{y_P + y_Q}{x_P + x_Q} = \frac{g^3 + g^{13}}{g^5 + g^9} = \frac{g^8}{g^6} = g^2$$

Second,  $x_R = \lambda^2 + \lambda + x_P + x_Q + a$  and  $y_R = \lambda(x_P + x_R) + x_R + y_P$

$$x_R = g^4 + g^2 + g^5 + g^9 + g^4 = g^3 \text{ and } y_R = g^2(g^5 + g^3) + g^3 + g^3 = g^{13}$$

Let  $P=(x_P, y_P)$ , compute the slope of the tangent line

$$\lambda = x_P + \frac{y_P}{x_P} = g^5 + \frac{g^3}{g^5} = g^5 + g^{-2} = g^5 + g^{13} = g^7.$$

Second,  $x_R = \lambda^2 + \lambda + a$  and  $y_R = x_P^2 + \lambda x_R + x_R$

$$x_R = g^{14} + g^7 + g^4 = 1 \text{ and } y_R = g^{10} + g^7 \cdot 1 + 1 = g^{13}$$

Recall the points on the curves:

$$g^0 = (0001), g^1 = (0010), g^2 = (0100), g^3 = (1000), g^4 = (0011),$$

$$g^5 = (0110), g^6 = (1100), g^7 = (1011), g^8 = (0101), g^9 = (1010),$$

$$g^{10} = (0111), g^{11} = (1110), g^{12} = (1111), g^{13} = (1101), g^{14} = (1001), g^{15} = (0001).$$

Let double a point  $P=(g^5, g^3) = (0110, 1000)$ . First, compute the slope of the tangent line

$$\lambda = x_p + \frac{y_p}{x_p}$$

Second,  $x_R = \lambda^2 + \lambda + a$  and  $y_R = x_p^2 + \lambda x_R + x_R$

3. In order to compute the slope we need to an inverse of  $x_p$ .

Given  $x_p = g^5 = (0110) = x^2 + x \text{ mod } f(x) = x^4 + x + 1 = (10011)$

The inverse of  $x_p$  is  $g^{-5}$  computed extended Euclidean algorithm.

An Extended Euclidean Algorithm

// To compute an inverse  $p(x) \text{ mod } \text{irreducible polynomial } f(x)$

$u(x)=0, v(x)=1, a(x)=p(x), b(x)=f(x); r(x)= b(x) \text{ mod } a(x)$

while  $r(x) > 0$  do

$$q(x) = b(x)/a(x); r(x) = b(x) - a(x)*q(x)$$

$$w(x) = u(x) - v(x)*q(x) \text{ mod } f(x)$$

$$b(x)=a(x), a(x) = r(x); u(x)=v(x), v(x)=w(x)$$

end//while

return  $v(x)$  //  $p(x)*v(x) = 1 \text{ mod } f(x)$ .

b	q	a	r	u	v	w
10011	111	0110	1	0	1	111
0110		1		1	111	

Let us compute  $b \text{ mod } a$

10011

110  $q=1$

01011

110  $q=11$

$$\begin{array}{ll} 0111 & \\ 110 & q=111 \\ 001 & r=1. \end{array}$$

$$\begin{aligned} w(x) &= u(x) - v(x)*q(x) \text{ mod } f(x) \\ &= 0 - 1*111 \text{ mod } 10011 = 111. \end{aligned}$$

From above

$$10011 = 110*111+1.$$

Let us compute

$$\begin{aligned} \lambda = x_p + \frac{y_p}{x_p} &= 0110 + 1000*111 = 0110 + 111000 = 111110 \text{ mod } 10011 \\ &= 111110 \\ &10011 \\ &= 11000 \\ &10011 \\ &= 1011 = g^7. \end{aligned}$$

Second,  $x_R = \lambda^2 + \lambda + a = g^{14} + g^7 + g^4 =$

$$\begin{array}{r} 1001 \\ 1011 \\ 0011 \\ \hline 0001 \end{array}$$

and  $y_R = x_p^2 + \lambda x_R + x_R = g^{10} + g^7 + g^0 =$

$$\begin{array}{r} 0111 \\ 1011 \\ 0001 \\ \hline 1101 \end{array}$$

$$2P = 2(g^5, g^3) = (g^0, g^{13}).$$

Let us build the generator table.  $P(g^5, g^3) = (0110, 1000)$ ,  $Q(g^9, g^{13}) = (1010, 1101)$ ,

Let the parameters  $a = g^4$  and  $b=1$ ,  $E: y^2 + xy = x^3 + ax^2 + b$  over an irreducible polynomial  $f(x) = x^4 + x + 1$ . Let  $x = g$ , then  $y^2 + gy = g^3 + g^6 + 1 = 1000 + 1100 + 0001 = 0101 = g^8$ .

$$y^2 + gy = g^8$$

$i$	$x_i = g^i$	$y_i$
0	0001	$g^{13} = 1101$
1	0010	
2	0100	

3	1000	
4	0011	
5	0110	$g^3 = 1000$
6	1100	
7	1011	
8	0101	
9	1010	$g^{13} = 1101$
10	0111	
11	1110	
12	1111	
13	1101	
14	1001	
15	0001	

## 2.7 Software and Hardware

Before this, the implementation of software codes to ECC written in C on Cent-OS 4.8. The code was written in NetBeans IDE 7.0.1 and as a compiler is gcc 3.4.6-11. To get better performance, use external libraries to deal with big numbers. In particular, the GNU multiple Precision Arithmetic Library, GMP 5.0.2, which contain code optimized for multiplication, division and other basic operations that have been used in dealing with big numbers. Other than that, the language that is used is Java programming language. As for this project which is Simulation of ECC on binary field will be using MATLAB programming language.

## 2.8 Conclusion

As a conclusion, in this chapter contain discussion on literature review that related with this project's topic. The example of existing system or research system are already mentioned in this chapter. It provide a clearer image what is the project is about. For the next chapter, methodology that will be used in this project will be discuss.

## CHAPTER 3

### METHODOLOGY

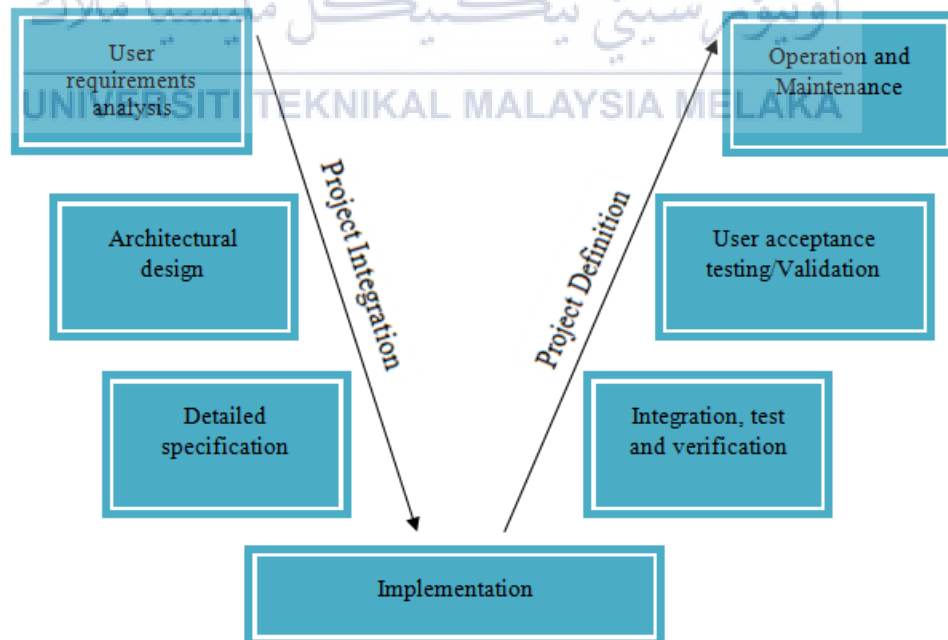
#### 3.1 Introduction

For this chapter which is project methodology which consists of the combination of the methods and technique of planning and delivery of the project. By referring the project milestone, it can be a guide to complete all the task required in this project. This project methodology will be used to develop the system. The entire project process will be controlled by this methodology to ensure the success for current technologies and business goals. In addition, the project methodology can helps developers to plan and as a guide for developers to be aware that what is the next step to complete all the process involved to develop the system and can finish it within the time given.

### 3.2 Methodology

The methodology used in this project is V-Shaped Model. The V-Shaped Model also known as Verification and Validation model. V-Shaped model is an extension version of the Waterfall model and every phase is depending on each other. To start the next phase, the previous phase need to be completed first. There are other System Development Life Cycle (SDLC) approaches that have been used which are Spiral Model, Agile Model, Iterative Model, Waterfall Model and Big Bang Model.

After reviewing the findings, it is time to analyze them and recommend the solutions for this project. Therefore, this topic of recommended solution is made up of several sub-topics which are the methodology used and the project schedule or milestones that need to describe later. So, while doing this chapter, it involves a lot on applying the knowledge in related subjects such as Software Development, Software Engineering, Project Management and etc. In order to ensure that the project management is effective, the structured methods which is the project methodology is needed.



**Figure 3.1 : V-Shaped Model**



The workflow of this model has been divided into seven phases which are user requirements analysis, architectural design, detail specification, implementation, integration, test and verification, user acceptance testing or validation and lastly is operation and maintenance as a guide to finish this project in a given time. All the activities in each phases will described as below.

i. Requirements

During this initial phase, the problem analysis and requirement analysis need to carried out. all the information that has been collected in the previous chapter which is from journal need to analyze first. For the activity of this phase, the problem that faced by the current system are needed to study first to analyze the problem. After all the problems have been defined and studied, determine all the requirements needed to ensure that the scopes and objectives of this project will be achieved.

ii. System design

As for this phase, the specifications of the system and what is the purpose of the system is needed to determine first. Next, determine the system requirements for business and technical to stay away from any limitation when doing this phase. The system design will have the understanding and detailing the complete hardware and software setup for this system.

At the end of this phase, sequence diagram and the interface of the system will be an output for this phase. Programming language of MATLAB will be used to design the interface and algorithm of the system

iii. Architectural design

During this phase, the detail on how the system will link up to all its various component will be drawn up. The system design will be broken down into modules that taking up different function. This also referred to as High-Level Design (HLD). Integration tests will be developed during this time.

iv. Module design

For this phase, it consists of the Low-Level Design (LLD) is used to explain on how the design of the system modules and the interfaces will

be functioned. Unit test will be created during this module design phase. Unit tests are an integral part of any development process and helps to eliminate maximum errors and errors at the initial stage.

- v. **Operation and implementation**  
For this phase, the design that have been created from the previous phase will be translated into coding. This project use the programming language which is MATLAB. This system need to carry out in the working environment. The activity of this phase is to see what and how the algorithm can successfully function. By the end of this phase, this system will be prepared for the next phase which is testing phase.
- vi. **Unit testing**  
For this phase, the complete system will be tested to make sure that there are no technical error occurred during final phase. After the system is finish, the system will be tested. It is end-to-end testing. This phase also should eliminate the majority of potential bugs and error in the system.
- vii. **Integration testing**  
During this phase, testing devised that have been done during the architectural design phase will be executed to ensure that the system functions correctly and to test the communication between the modules within the system.
- viii. **System testing**  
System testing is interact directly with the system design phase. It also check the entire system functionality whether function correctly or not. Software and hardware compatibility issues can be detect during the execution of this system testing.
- ix. **Maintenance**  
The final phase is maintenance. After the system have been completed and tested, supervisor and evaluators will conduct the user acceptance test. If the system is accepted, the simulation will be published and established by the users. further maintenance will be conduct in the future to ensure

that the bugs is fixed and to make sure that the system can perform and run well without any technical errors.

### 3.3 Project Milestone

Gantt chart has the major benefits to schedule the activities that need to be done before the due date of a project. Gantt charts have been created to ensure that the project are on the right track, providing a visual timeline for the start and finish certain tasks. By providing a visual representation of the key events and other important dates, this chart is considered to offer an easier method to understand and remember to maintain service-based timescales and deliverables either tracked daily, weekly, monthly or yearly. Each of the phase has its own activities that need to be done. The table below shows the detail of the activities and date.

**Table 3.1 : Gantt Chart**

Months & Week	February				March				April				May				June				July				August			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Requirements																												
Design																												
Implementation																												
Testing																												
Maintenance																												

**Table 3.2 : Milestone Activities**

Activities	Duration
<p><b>Requirements Phase</b></p> <ul style="list-style-type: none"> <li>I. Problem statement and objectives</li> <li>II. Project scope</li> <li>III. Make a schedule</li> <li>IV. Review current system</li> <li>V. Research</li> </ul>	<p>20 February 2017 – 12 March 2017</p>
<p><b>Design Phase</b></p> <ul style="list-style-type: none"> <li>I. Logical design</li> <li>II. System Interface</li> </ul>	<p>13 March 2017 – 7 May 2017</p>
<p><b>Implementation Phase</b></p> <ul style="list-style-type: none"> <li>I. Algorithm</li> </ul>	<p>8 May 2017 – 25 June 2017</p>
<p><b>Testing Phase</b></p> <ul style="list-style-type: none"> <li>I. Testing on the system</li> </ul>	<p>26 June 2017 – 1 August 2017</p>
<p><b>Maintenance Phase</b></p> <ul style="list-style-type: none"> <li>I. Fixing the bugs on the system</li> </ul>	<p>1 August 2017 – 31 August 2017</p>

### 3.4 Finite Fields

This is the brief introduction about finite field. Addition and multiplication is the operations of two binary on  $F$  that combine with a set of finite element  $F$  to form a finite field, which meets with the some of the arithmetic characteristics. The number of elements in the field is the order of a finite field. There is finite field of order  $q$  if and only if the prime power is  $q$ . If a prime power is  $q$ , then basically only one finite field of order  $q$ ; this finite field is marked with  $F_q$ . In the next subtopic, the elements and finite field  $F_p$  operations and the elements and finite field  $F_2^m$

operations will be discussed, along with the methods to represent field elements which is polynomial basis representations. (Johnson, D., Menezes, A., & Vanstone, S. 2001)

### 3.4.1 Finite Field $F_p$

The  $p$  is denoted as prime number. Finite field  $F_p$  is called a prime field is comprised of set of integers  $(0, 1, 2, \dots, p-1)$  with following arithmetic operations.

#### Addition

If  $a, b \in F_p$ , then  $a+b=r$ , where  $r$  is remainder when  $a+b$  is divided by  $p$  and  $0 \leq r \leq p-1$ . This is called as addition modulo  $p$ .

Let us take an example of finite field =  $F_{23}$ , element  $F_{23} = \{0, 1, 2, \dots, 23-1\} = \{0, 1, 2, \dots, 22\}$ .

$$a=12, b=20, a+b=r.$$

$$a+b=12+20=9$$

$$\text{where, } 12+20=32$$

$$\text{then, } 32 \bmod 23 = 9 \text{ remainder } 9$$

$$0 \leq 9 \leq 22, \text{ where } r=9$$

#### Multiplication

If  $a, b \in F_p$ , then,  $a \cdot b = s$  where  $s$  is remainder when  $a \cdot b$  is divided by  $p$  and  $0 \leq s \leq p-1$

Let us take an example of finite field =  $F_{23}$ , element  $F_{23} = \{0, 1, 2, \dots, 23-1\} = \{0, 1, 2, \dots, 22\}$ .

$$a = 8, b = 9, a \cdot b = s$$

$$a \cdot b = 8 \cdot 9 = 3$$

$$\text{where, } 8 \cdot 9 = 72$$

$$\text{then, } 72 \bmod 23 = 3 \text{ remainder } 3$$

$$0 \leq 3 \leq 22, \text{ where } s = 3$$

## Inversion

If  $a$  is non zero element in  $F_p$ , inverse of  $a$  modulo  $p$ , denoted at  $a^{-1}$  is the unique integer and  $c \in F_p$  which  $a \cdot c = 1$ .

Let us take an example of finite field =  $F_{23}$ , element  $F_{23} = \{0, 1, 2, \dots, 23-1\} = \{0, 1, 2, \dots, 22\}$ .

$$a = 8, a^{-1} = 8^{-1}$$

$$8^{-1} = 3 \pmod{23}$$

$$\text{where, } 8 \cdot 3 = 1$$

then,  $8 \pmod{23} = 3$ .

### 3.4.2 Finite Field $F_{2^m}$

Binary field  $F_{2^m}$ , where  $m$  is a positive integer starting from 160, 256, 384 and 512. Two main advantages of binary finite field are the bit additions are performed by mod 2 and represented in hardware by XOR gates which is no carry chain is needed. Other than that, the bit multiplications are represented in hardware by AND gates. The  $F_{2^m}$  elements can be viewed as vectors dimension  $m$  where each of bit can only take values of "0" or "1". Finite fields of order  $2^m$  are called binary fields or characteristic-two finite fields. One way to construct  $F_{2^m}$  is to use a polynomial basis representation. Here, the elements of  $F_{2^m}$  are the binary polynomials (polynomials whose coefficients are in the field  $F_2 = \{0,1\}$ ) of degree at most  $m-1$ :

$$F_{2^m} = p(z) = \{a_{m-1}z^{m-1} + a_{m-2}z^{m-2} + \dots + a_2z^2 + a_1z^1 + a_0z^0 : a_i \in \{0,1\}\}$$

An irreducible binary polynomial  $f(z)$  of degree  $m$  is chosen. Irreducibility of  $f(z)$  means that  $f(z)$  cannot be factored as a product of binary polynomials each of degree less than  $m$ . Addition of field elements is the usual addition of polynomials, with coefficient arithmetic performed modulo 2. Multiplication of field elements is performed modulo the reduction polynomial  $f(z)$ .

Let  $f(z) = f_{m-1}z^{m-1} + \dots + f_2z^2 + f_1z^1 + f_0z^0$  be an irreducible polynomial of degree  $m$  over  $F_2$ .

Let us take the element of  $F_{2^4}$  are the 16 binary polynomials of degree at most 3

$$g^0 = 0 = 0000,$$

$$g^1 = 1 = 0001,$$

$$g^2 = z = 0010,$$

$$g^3 = z + 1 = 0011,$$

$$g^4 = z^2 = 0100,$$

$$g^5 = z^2 + 1 = 0101,$$

$$g^6 = z^2 + z = 0110,$$

$$g^7 = z^2 + z + 1 = 0111,$$

$$g^8 = z^3 = 1000,$$

$$g^9 = z^3 + 1 = 1001,$$

$$g^{10} = z^3 + z = 1010,$$

$$g^{11} = z^3 + z + 1 = 1011,$$

$$g^{12} = z^3 + z^2 = 1100,$$

$$g^{13} = z^3 + z^2 + 1 = 1101,$$

$$g^{14} = z^3 + z^2 + z = 1110,$$

$$g^{15} = z^3 + z^2 + z + 1 = 1111.$$

Next, let us take some examples of arithmetic operations in  $F_{2^4}$  with

reduction polynomial  $f(z) = z^4 + z + 1$ .

### Addition

The addition of two polynomials is bitwise exclusive-or operation. Let

$$g^i = (g_{m-1} \dots g_2 g_1 g_0) \text{ and } g^j = (g_{m-1} \dots g_2 g_1 g_0), \text{ then}$$

$$c = g^i + g^j = (g_{m-1} \dots g_2 g_1 g_0) \oplus (g_{m-1} \dots g_2 g_1 g_0) = (g_{m-1} \dots g_2 g_1 g_0).$$

$$\text{Let } g^{13} = z^3 + z^2 + 1 = 1101_2 \text{ and } g^9 = z^3 + 1 = 1001_2$$

$$1101$$

$$\underline{1001}$$

$$0100$$

$$c = g^{13} + g^9 = (g_{m-1} \cdots g_2 g_1 g_0) \oplus (g_{m-1} \cdots g_2 g_1 g_0) = z^2 = 0100_2$$

### Multiplication

$g^{13} \cdot g^9 = d = (d_{m-1} \cdots d_2 d_1 d_0)$  where  $d(z) = d_{m-1}z^{m-1} + \cdots + d_2z^2 + d_1z^1 + d_0z^0$  is

the remainder of

$$g^{13}(z) \cdot g^9(z) = (g_{m-1}z^{m-1} + \cdots + g_2z^2 + g_1z^1 + g_0z^0) \cdot (g_{m-1}z^{m-1} + \cdots + g_2z^2 + b_1z^1 + g_0z^0)$$

modulo irreducible polynomial  $f(z) = f_{m-1}z^{m-1} + \cdots + f_2z^2 + f_1z^1 + f_0z^0$ .

Let  $g^{13} = z^3 + z^2 + 1 = 1101_2$  and  $g^9 = z^3 + 1 = 1001_2$ ,  $f(z) = z^4 + z + 1 = 10011_2$

$$g^{13} \cdot g^9 \text{ mod } 10011_2 = 1101_2 \cdot 1001_2 \text{ mod } 10011_2,$$

$$1101_2$$

$$1001_2$$

$$1102101_2 \text{ mod } 10011_2,$$

$$10011_2,$$

$$0101001_2$$

$$10011_2,$$

$$1102101_2 \text{ mod } 10011_2,$$

$$10011_2,$$

$$0101001_2$$

$$10011_2,$$

$$001111_2 = g^{15} = (1111).$$

$$g^{13} \cdot g^9 = 1101_2 \cdot 1001_2 \text{ mod } 10011_2 = g^{15} = 1111_2$$

### Inversion

$g^{13}(z) = g_{m-1}z^{m-1} + \cdots + g_2z^2 + g_1z^1 + g_0z^0$ , denoted as  $g^{-1}$ , is the unique element

$c \in F_{2^m}$  such that  $g(z) \cdot c(z) \equiv 1 \text{ mod } f(z)$

$$(g_{m-1}z^{m-1} + \cdots + g_2z^2 + g_1z^1 + g_0z^0) \cdot (c_{m-1}z^{m-1} + \cdots + c_2z^2 + c_1z^1 + c_0z^0) \\ \equiv 1 \text{ mod } f_{m-1}z^{m-1} + \cdots + f_2z^2 + f_1z^1 + f_0z^0$$

Let us do an example of  $g^{13} = 1101_2$  and  $g^9 = 1001_2 \in F_{2^4}$



Let us try an inversion via the extended Euclidean Algorithm. To compute the inverse of polynomial  $g(z)^{13} = z^3 + z^2 + 1 = 1101_2 \pmod{10011}$

Suppose we want to compute an inverse  $g^{-1} = 1101^{-1} \pmod{10011}$ .

**Table 3.3 : Igcd**

$I$					$v_i$
-2	igcd				0
-1	b=	a*	q+	R	1
0	10011	1101	10	1001	-10
1	1101	1001	1	100	11
2	1001	100	10	1	100

Let us see the division in binary,

$$v = v_i - v_{i-2} * q_i$$

$$i=0,$$

$$10011 \pmod{1101}$$

$$\begin{array}{r} 10011 \\ \oplus 1101 \quad q: 1 \\ \hline 1001 \quad q: 10 \end{array}$$

$$v_i = v_{i-2} - v_{i-1} * q_i \pmod{f} = 0 - 1 \cdot 10 = -10.$$

$$i=1,$$

$$1101 \pmod{1001}$$

$$\begin{array}{r} 1101 \\ \oplus 1001 \\ \hline 100 \quad q: 1 \end{array}$$

$$v_i = v_{i-2} - v_{i-1} * q_i \pmod{f} = 1 - (-10) \cdot 1 = 1 + 10 = 11.$$

$$i=2,$$

$$1001 \pmod{100}$$

$$\begin{array}{r} 1001 \\ \oplus 100 \quad q: 1 \\ \hline 1 \quad q: 10 \end{array}$$

$$v_i = v_{i-2} - v_{i-1} * q_i \text{ mod } f = -10 - 11 \cdot 10 = -10 - 110 = 100.$$

Now, we get the answer as

$$g^{-1}(z) = z^2 = 0100$$

Let us reconfirm the answer :  $g^{-1}(z) \cdot g(z) \equiv 1 \text{ mod } f(z)$

$$1101 \cdot 0100 = 110100 \text{ mod } 10011$$

$$\begin{array}{r} 110100 \\ \oplus 10011 \\ \hline 10010 \\ \oplus 10011 \\ \hline 1 \end{array} \quad \begin{array}{l} q : 1 \\ q : 10 \\ q : 11 \end{array}$$



### 3.5 Recommended Finite Fields

There are 10 recommended finite fields by NIST which are :

i. Prime fields  $F_p$  :

$$p = 2^{192} - 2^{64} - 1, p = 2^{224} - 2^{96} + 1, p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1, p = 2^{384} - 2^{128} - 2^{96} + 2^{32} - 1, p = 2^{521} - 1$$

ii. Binary fields  $F_{2^m}$  :

$$F_{2^{163}}, F_{2^{233}}, F_{2^{283}}, F_{2^{409}}, F_{2^{571}}$$

### 3.6 Conclusion

As a conclusion, the project methodology used in this project is V-Shaped Model methodology. The project schedule was created in the form of a Gantt Chart so that each of the task can be done on the time provided. The design phase of Simulation of ECC on binary field will be discussed later on the next chapter.

## CHAPTER 4

### DESIGN

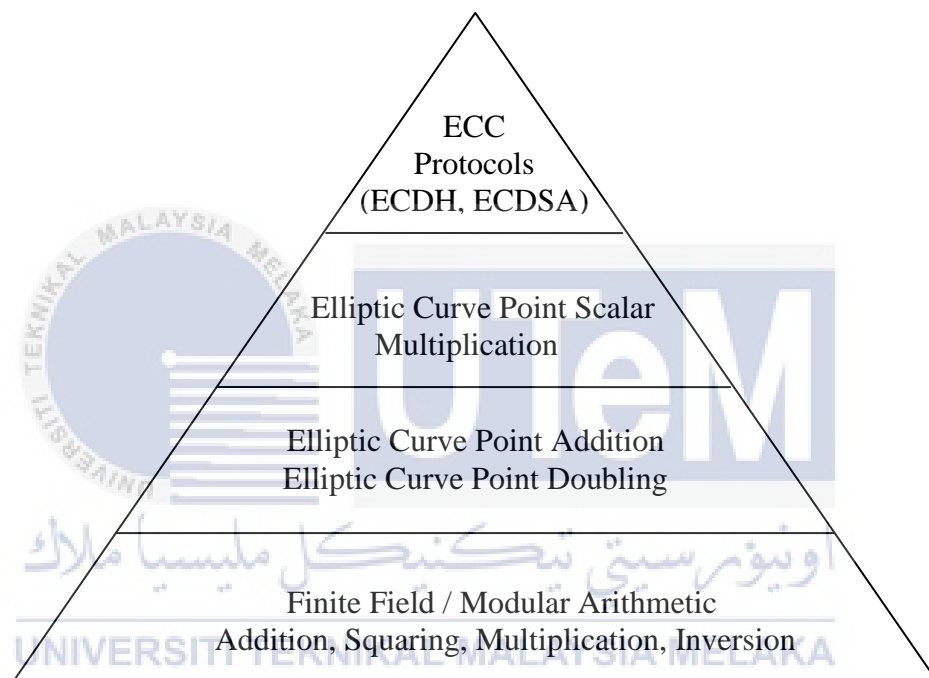
#### 4.1 Introduction

Simulation of ECC on binary field is a system to compute the three operation involve in binary field which are addition, multiplication and inversion. One way to construct binary field is to use a polynomial basis representation. The elements of binary field are the binary polynomials which is the polynomials with coefficient arithmetic performed modulo 2. ECC algorithm is the algorithm that need to use in this system. The ECC algorithm consists of four parameter which are  $m$ ,  $a$ ,  $b$  and  $f(x)$ .

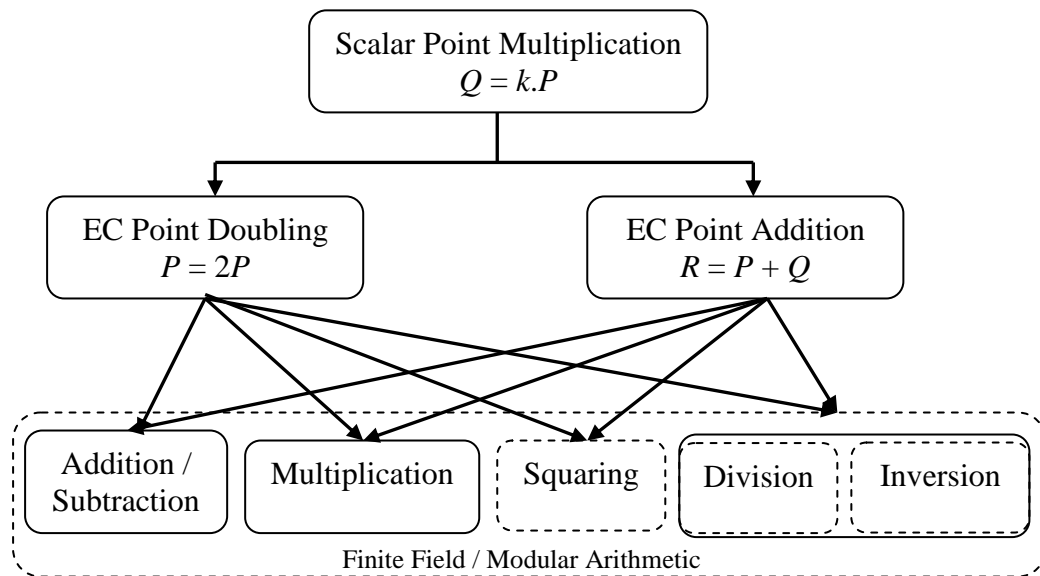


## 4.2 System Architecture

In the implementation of ECC cryptosystem, it involves the hierarchy of the computations which is illustrated in a pyramid which consists of four levels of operations. Finite field or modular arithmetic is the basis of the pyramid because it is the base of building blocks of the elliptic curve point addition and point doubling. While scalar multiplication (SM) is conducted by redoing the point addition and point doubling operations and used by all ECC cryptographic protocols.



**Figure 4.1 : ECC Operational Pyramid**



**Figure 4.2 : Arithmetic Dependency**

As for this system in the project, the simulation of ECC on binary field cover all of those operation and point representation as in the figure above. There are four operation in the ECC over binary field which are addition/subtraction, multiplication, squaring and inversion. The operation of division is also a part of the operation of inversion which means the operation of division is calculated first to compute the operation of inversion.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

#### 4.2.1 Interface

In this subtopic, the design of the interface will be provided. User can interact with this system and will be able to see what is the input and output of this system and they may know the step on how to use the system.

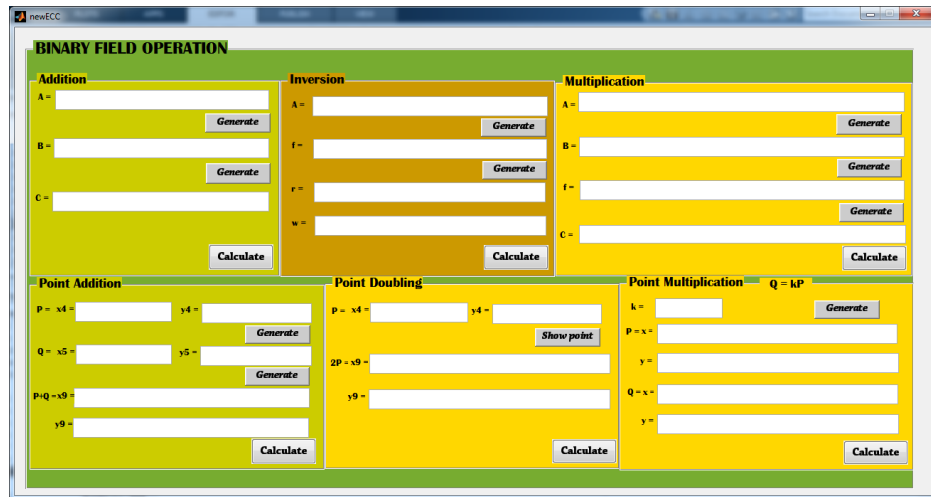


Figure 4.3 : Main Menu Interface

#### 4.2.2 Addition Interface

Addition is the first operation in this system which is the number is a random number that will be generate when the button is being pushed. The output will be displayed here.

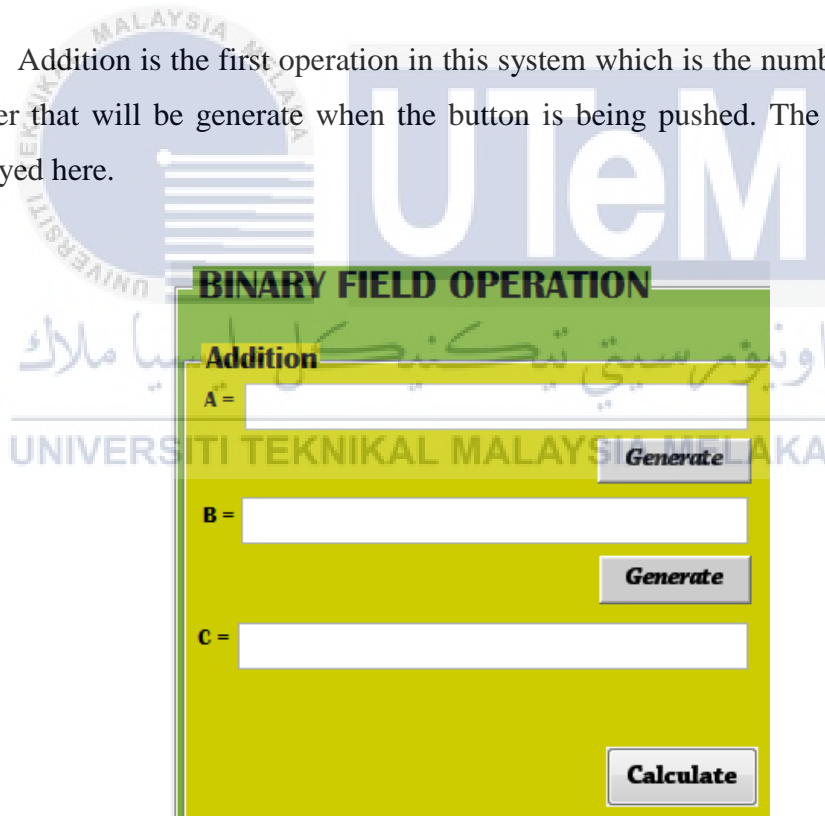


Figure 4.4 : Addition Interface

### 4.2.3 Multiplication Interface

The second operation is multiplication which is number will be generate randomly when the button is pushed and the output from will be displayed here.

**Multiplication**

A =  **Generate**

B =  **Generate**

f =  **Generate**

C =  **Calculate**

Figure 4.5 : Multiplication Interface

### 4.2.4 Inversion Interface

The last operation is inversion which the number also will be generate randomly after the pushed the button and the output will be displayed here.

**Inversion**

A =  **Generate**

f =  **Generate**

r =

w =  **Calculate**

Figure 4.6 : Inversion Interface

#### 4.2.5 Point Addition Interface

The first point representation is point addition which the point of  $x$  and  $y$  will be randomly generate. This point representation involve the operation of addition, multiplication and inversion. The output will be displayed here.

**Point Addition**

P =  $x_4$  =        $y_4$  =       **Generate**

Q =  $x_5$  =        $y_5$  =       **Generate**

P+Q =  $x_9$  =       **Calculate**

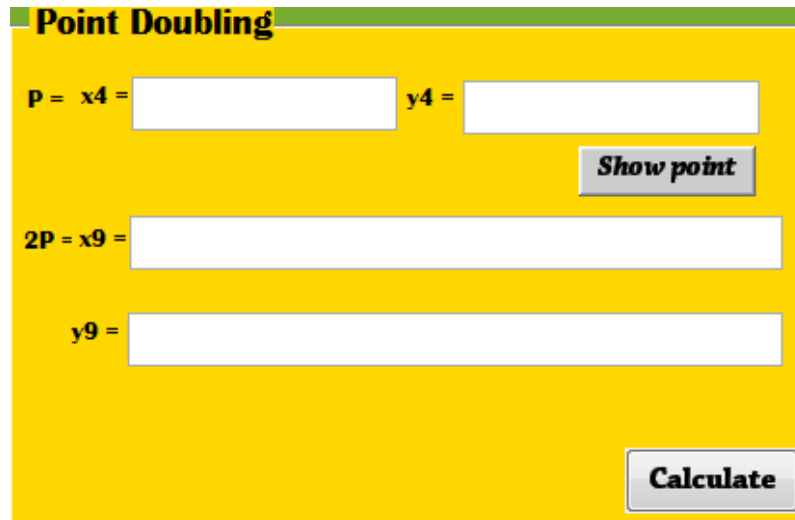
$y_9$  =

Figure 4.7 : Point Addition Interface

#### 4.2.6 Point Doubling Interface

The second point representation is point doubling which the point of  $x$  and  $y$  will be randomly generate from the previous point addition. This point representation involve the operation of addition, multiplication and inversion. The output will be displayed here.





**Point Doubling**

$P = x4 =$    $y4 =$

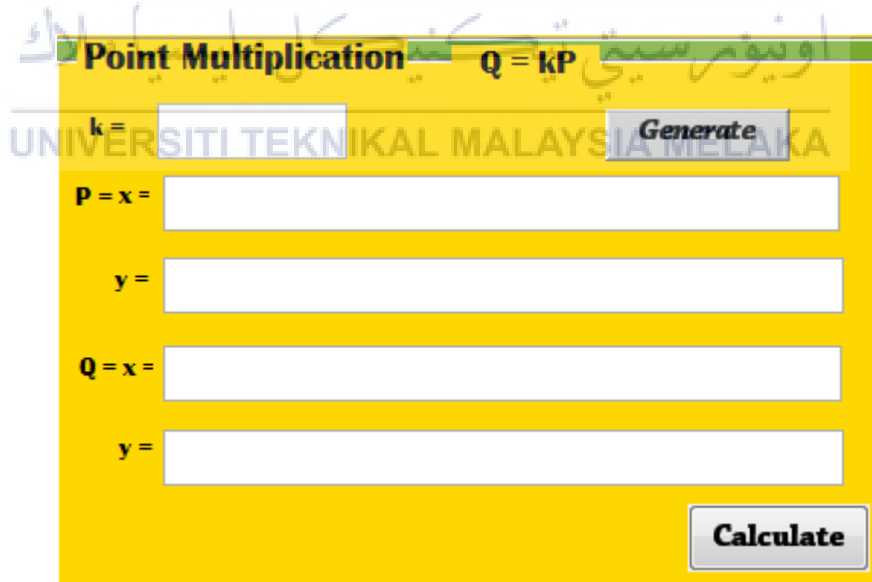
$2P = x9 =$

$y9 =$

Figure 4.8 : Point Doubling Interface

#### 4.2.7 Point Multiplication Interface

The last point representation is point multiplication which the point of  $x$  and  $y$  will be randomly generate from the previous point addition.  $k$  is needed to repeat the process.  $k$  will be generate randomly. This point representation involve the operation of addition, multiplication and inversion. The output will be displayed here.



**Point Multiplication**  $Q = kP$

$k =$

$P = x =$

$y =$

$Q = x =$

$y =$

Figure 4.9 : Pont Multiplication Interface

### 4.3 Logical Design

Flowchart below shows the process of input and output of the Simulation of ECC on binary field.

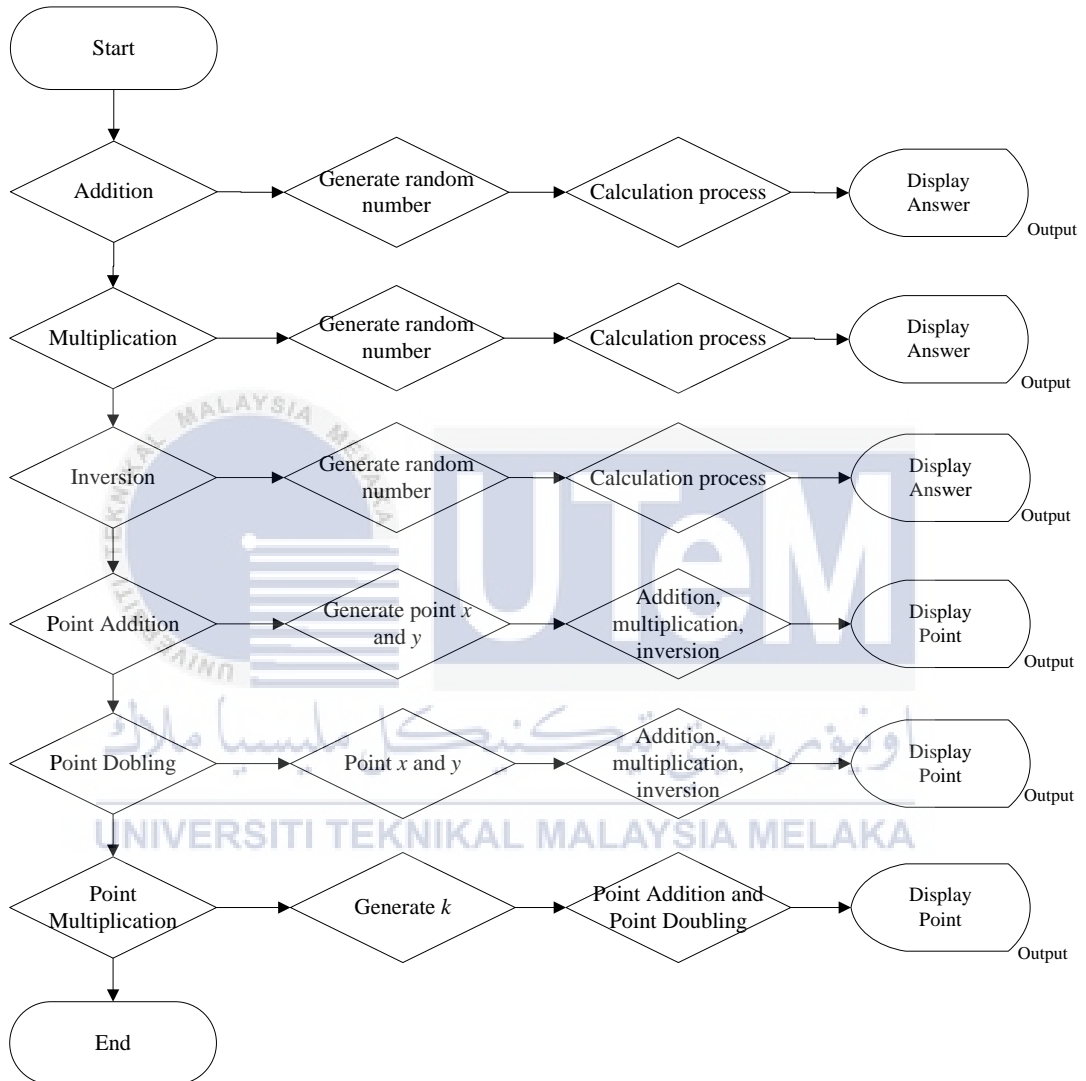


Figure 4.10 : Flowchart

#### 4.4 Possible Scenario

This subtopic will be shown the algorithm that has been produced in MATLAB based on the tutorial reference.

```
% ECC modulo an irreducible polynomial for addition operation

function c = Add(a,b)

%Assume the inputs are polynomial a(x) and b(x) of the same size
c=bitxor(a,b);

% ECC modulo an irreducible polynomial for multiplication
operation
function c = Multiply(a,b)

% Assume the inputs are normally a(x) > b(x)
% bitget starts from LSB

n1=ceil(log2(a));n2=ceil(log2(b));
if n1==0 || n2==0,
    c=0;
else
    if n1 < n2,
        temp=a; a=b;b=temp;
        size=n1; n1=n2; n2=size;
    end
    a=bitshift(a,n2-1)
    c=a
    a=bitshift(a,-1)
    for i=n2-1:-1:1,
        bitget(b,i)
        ifbitget(b,i)
            c=Add(a,c)

        end
        a=bitshift(a,-1)
    end
end
end
end
```

```

% to calculate the operation of division

function [q,r2] = division(f2,a2)

m=length(a2)-1;
n=length(f2)-1;
r2=zeros(1,n+1);
q=[];
r2=f2;

for i=1:n-m+1,
s=zeros(1,n+1);
s(i:m+i)=a2
if r2(i)==1,
    q=[q 1]
    r2=bitxor(r2,s)
else
    q=[q 0]
end
end

% to calculate the operation of inversion after compute division

function [q,w2] = divw(f2,x)

m=length(x)-1;
n=length(f2)-1;
w2=zeros(1,n+1);
q=[];
w2=f2;

for i=1:n-m+1,
s=zeros(1,n+1);
s(i:m+i)=x;
if w2(i)==1,
    q=[q 1];
    w2=bitxor(w2,s);
else
    q=[q 0];
end
end

```

## 4.5 Conclusion

The significance of this chapter is to design the system of this project. This is because the design of the system will help to develop the system in systematic way. It also includes the step to design about the network or system architecture, interfaces, logical design which includes flowchart of the system. A layered architecture is used to explain the different function on each level of operation. The logical design are being defined by using the activity diagram. The next chapter is about the implementation phase of this project. As for that, the results from this chapter are very significant in order to proceed to the next chapter.



## CHAPTER 5

### IMPLEMENTATION

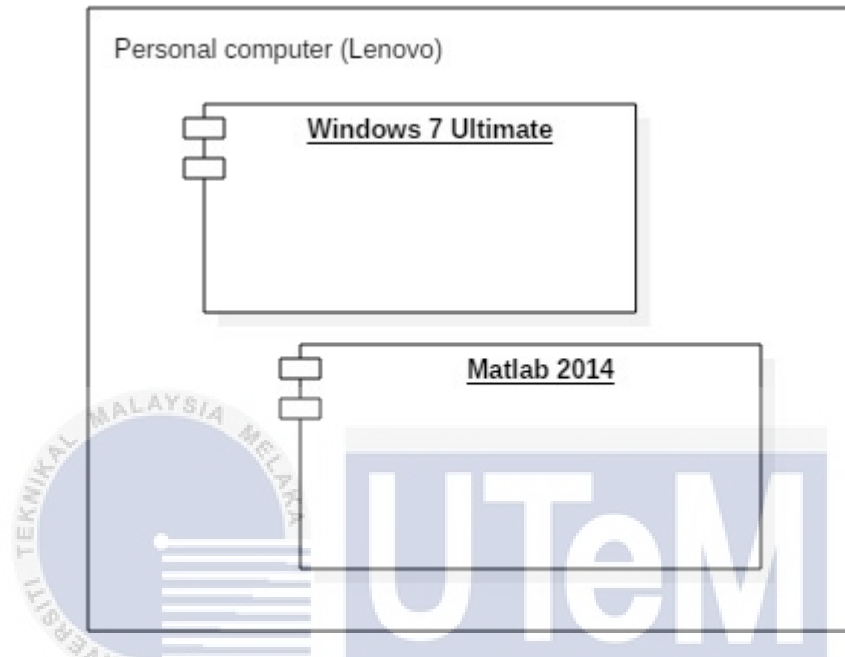
#### 5.1 Introduction

For this implementation chapter, it is a phase of where the manual calculation of ECC over binary field which covered the operation of addition, multiplication and inversion also point representation which are point addition and point doubling are translated into the code of Matlab, This chapter consists of design development, software installation and the configuration and implementation of the system.

In the topic of design development, deployment diagram is used to deploy the system based on the software used which is MATLAB. Besides that, in the topic of the software installation, the step by step on how to install the MATLAB will be discussed later. Finally, the topic of configuration and implementation will be describing about the progress in the developing of the simulation of ECC over binary field.

## 5.2 Software Development Environment Setup

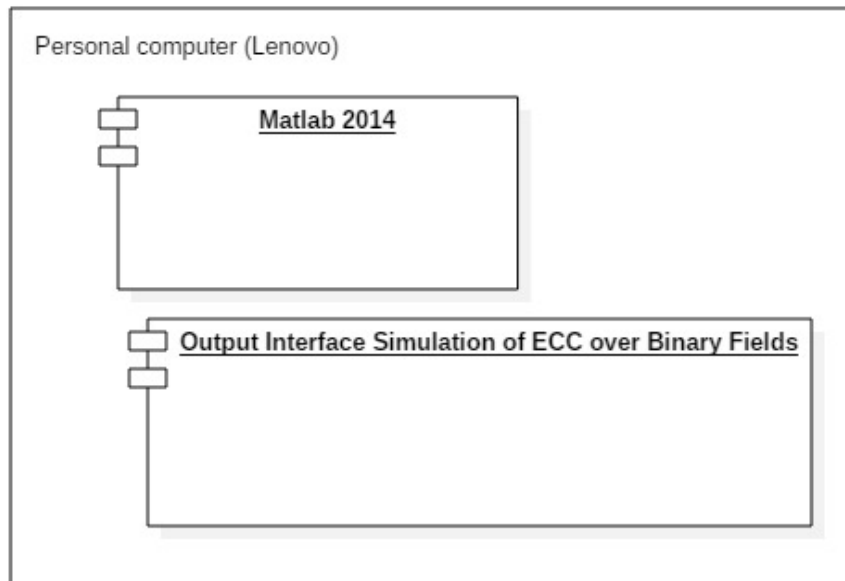
This topic will be discussed about the architecture of software and hardware development environment setup. The software architecture environment setup of simulation of ECC over binary field is shown as in the Figure 5.1 below.



**Figure 5.1 : Software Development Environment Setup Architecture**

The figure above shows that the hardware required to develop this system is only the personal computer. For this project, a laptop with the brand Lenovo, which is my own laptop, is used to develop the simulation as it is easy to be carried from one place to another as it is very lightweight. The operating system used in this project is Windows 7 Ultimate 64-bit. Lastly, the software used for this project is MATLAB 2014a, which consists of a GUI by using GUIDE, a programming language, and other mathematical algorithms. After producing the code in the function, the file will be saved as a .m extension file.

After the environment setup, the next step is to plan how the system will work. The only hardware used in this project is one personal computer that is a standalone computer. MATLAB software needs to run to show the interface of the system that has been developed. Figure 5.2 shows the deployment diagram of the simulation of ECC over binary field functionality.



**Figure 5.2 : Deployment Diagram**

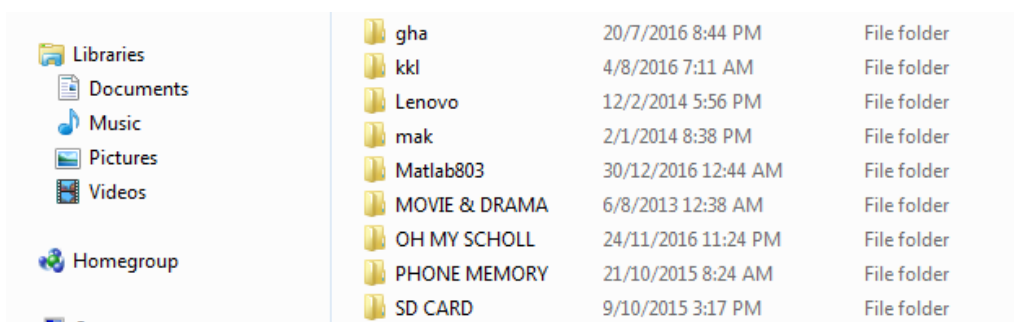
### 5.3 Software Configuration Management

In this topic, the step by step on how to install the MATLAB 2014a will be explained in detail as it is the software that is used in this project.

#### 5.3.1 MATLAB Installation

The step by step of the installation of MATLAB 2014a will be shown below.

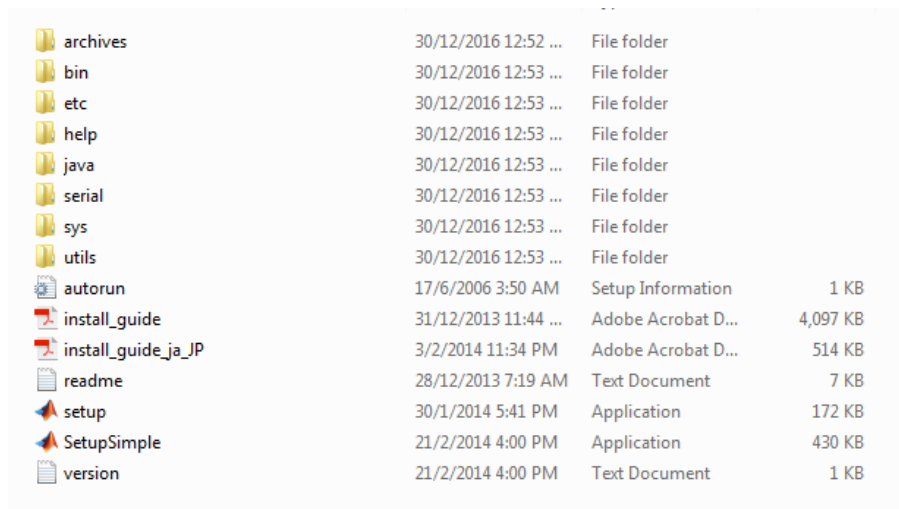
Step 1 : Open the Matlab803 file after the supervisor provide this software because it is legal software.



**Figure 5.3 : MATLAB file in the Documents Folder**

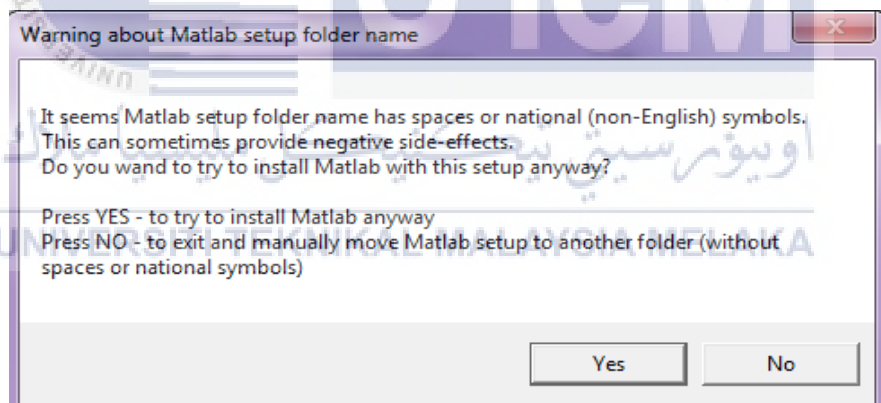


Step 2 : Click on SetupSimple



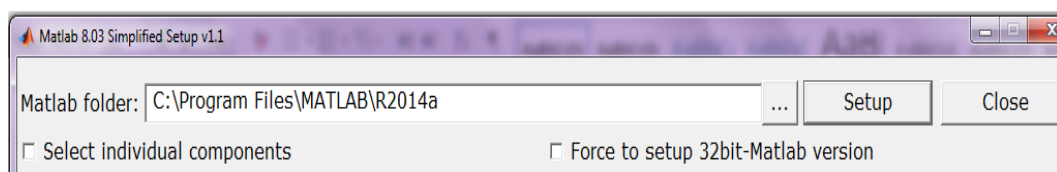
**Figure 5.4 : Matlab803 File**

Step 3 : After click SetupSimple, there will be pop up windows that is the warning about Matlab setup folder name. Click Yes to continue.



**Figure 5.5 : Warning Message MATLAB**

Step 4 : After that, there is another pop up windows for Matlab 8.03 Simplified Setup. Click the Setup to choose the component that needed only.



**Figure 5.6 : Setup**

Step 5 : Tick Select individual components and choose the component needed.

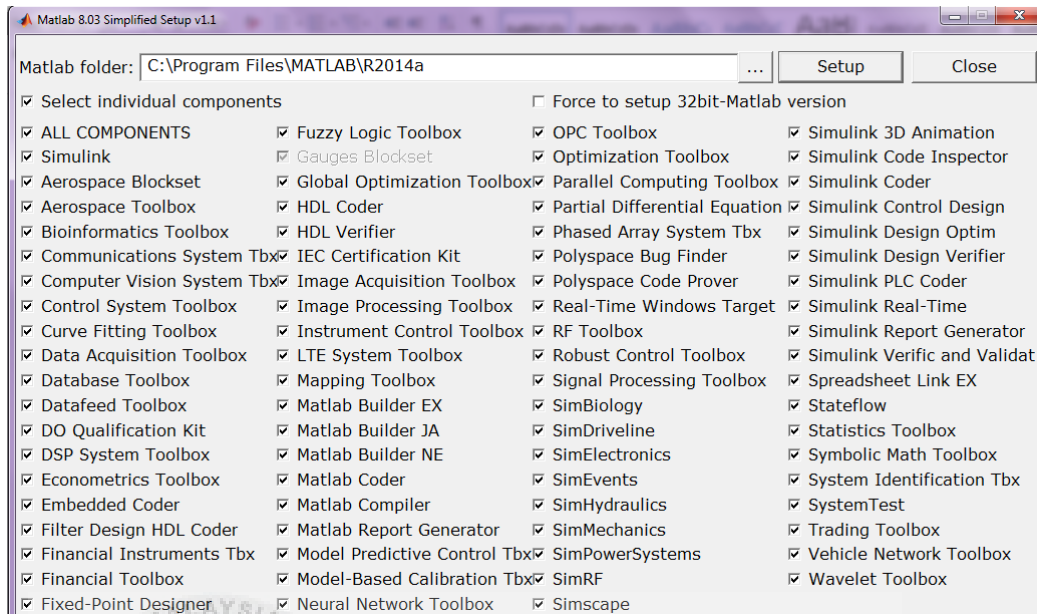


Figure 5.7 : Select Component

Step 6 : After the process of installing have been finished, there will be icon like this on the Desktop.

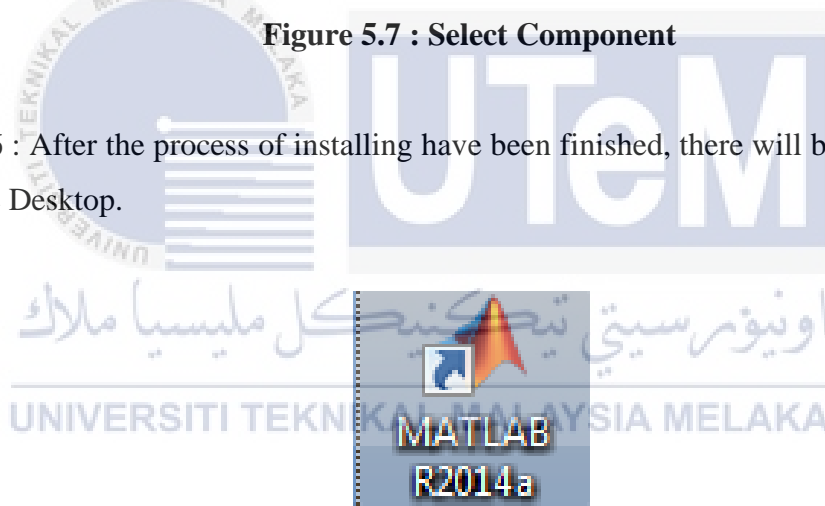


Figure 5.8 : Matlab Icon

Step 7 : Open Matlab and the software is ready to be used.

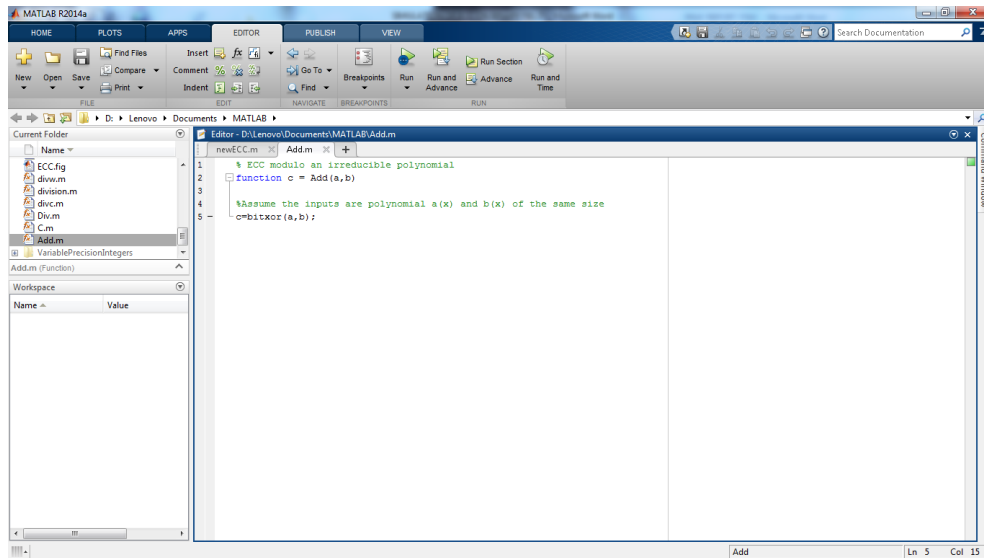


Figure 5.9 : Matlab Main Page

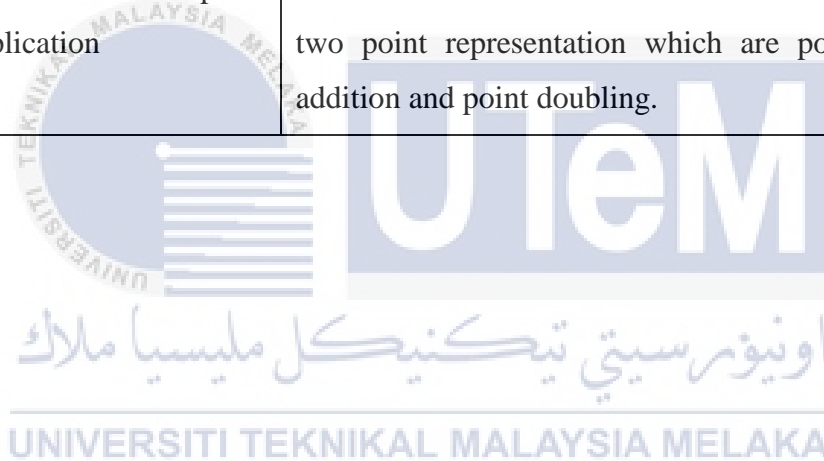
#### 5.4 Implementation Status

The task of implementation status is the task that describes about the development for each status of Simulation of ECC over binary field. Table 5.1 as show below will show the implementation of this simulation.

Table 5.1 : Implementation Status

Module	Description	Duration
Generate random number with specific size	This module provided random number that will be generate when the button is being pushed. User does not need to insert or enter the value by themselves.	2 Days
Calculate the operation of addition	This module provided calculation of the operation addition which is the by doing the exclusive-or between two numbers.	3 Days
Calculate the operation of multiplication	This module provided the calculation of the operation multiplication which includes the convolution and division between two numbers.	2 Weeks

Calculate the operation of inversion	This module provided the calculation of the inversion which includes the operation of exclusive-or, division and convolution.	2 Weeks
Calculate the point representation which is point addition by using the formula of Elliptic Curve	This module will includes the operation of addition, multiplication and inversion by using point $(x,y)$ .	2 Days
Calculate the point representation which is point doubling by using the formula of Elliptic Curve	This module will includes the operation of addition, multiplication and inversion by using point $(x,y)$ .	2 Days
Calculate the point multiplication	This module will includes the calculation of two point representation which are point addition and point doubling.	1 Week



## 5.5 Code of the System

In this part the code of the system will be shown using the programming language of MATLAB

### pointaddition.m

```
function [x9,y9] = pointaddition(x4,y4,x5,y5)

%to find x9
d=bitxor(y4,y5); %y4+y5
e=bitxor(x4,x5); %x4+x5
f=[1 0 0 0 1 1];
[q,g] = pointadditiondivx9(d,e); % (y4+y5) / (x4+x5) = w3

u=zeros(1,m+1);
v=zeros(1,m+1);
v(m+1)=1;
v2=conv(v,q);
u(numel(v2)) = 0;
x3=bitxor(u,v2);

[q,w3]=painversex9(f,x3);

k=conv(w3,w3); % (w3)^2
k=mod(k,2);
j=f(4:6);
f=bitxor(f(1:3),j);
[p,h] = pointadditionsqux9(f,k);

w3(numel(h)) = 0;
x=bitxor(h,w3); % (w2)^2 + w2 + x4 + x5 + a4 = x9
y=bitxor(x4,x5);
y(numel(x)) = 0;
z=bitxor(x,y);
a4=[0 0 0 1];
a4(numel(z)) = 0;
x9=bitxor(z,a4);

%to find y9 %w2*(x4+x9) + (x9+y4) = y9
x4(numel(x9)) = 0;
l=bitxor(x4,x9); %x4+x9=1
y4(numel(x9)) = 0;
m=bitxor(x9,y4); %x9+x4=m

f=[1 0 0 0 1 1];
a=conv(w3,l); % (y4+y5/x4+x5) (x4+x9) = w3*1
a=mod(a,2);
b=f(4:6);
f=bitxor(f(1:3),b);
[p,n] = pointadditionmuly9(f,a);
n(numel(m)) = 0;
y9=bitxor(n,m);
```

## pointdoubling.m

```
function [x9d,y9d] = pointdoubling(x4,y4,f)

%to find x9d
r=conv(x4,x4);          %x4^2
r=mod(r,2);
s=f(4:6);
f=bitxor(f(1:3),s);
[p,t] = pointdoublingsqux4(f,r);

b=[ 0 0 0 1];
b(numel(t)) = 0;
[p,q] = pointdoublingdivx4(b,t);      %b/x4^2

u=zeros(1,m+1);
v=zeros(1,m+1);
v(m+1)=1;
v2=conv(v,p);
u(numel(v2)) = 0;
x6=bitxor(u,v2);

f(numel(x6)) = 0
[q,w4]=pdinversex9(f,x6);

t(numel(w4)) = 0
x9d=bitxor(t,w4);          % (x4^2) + (b/x4^2)

%to find y9d
y4(numel(x4))= 0
[p,u] = pointdoublingdivy9d(y4,x4);  %y4/x4

u5=zeros(1,m+1);
v5=zeros(1,m+1);
v5(m+1)=1;
v2=conv(v5,p);
u5(numel(v2)) = 0;
x7=bitxor(u5,v2);

[q,w5]=pdinversey9(f,x7);

x4(numel(w5)) = 0
v=bitxor(x4,w5);          % (x4 + (y4/x4))

e=conv(v,x9d);          % (x4 + (y4/x4)) * x9d
e=mod(e,2);
e1=f(4:6);
f=bitxor(f(1:3),e1);
[p,o] = pointdoublingmulx9d(f,e);

o(numel(x9d)) = 0          % (x4 + (y4/x4)) * x9d + x9d
y9d=bitxor(o,x9d)
```

### division.m

```
function [q,r2] = division(f2,a2)

m=length(a2)-1;
n=length(f2)-1;
r2=zeros(1,n+1);
q=[];
r2=f2;

for i=1:n-m+1,
s=zeros(1,n+1);
s(i:m+i)=a2
if r2(i)==1,
    q=[q 1]
    r2=bitxor(r2,s)
else
    q=[q 0]
end
end
```

### addition.m

```
c1 = bitxor(a1,b1)
```

### multiplication.m

```
y=conv(a3,b3)
y=mod(y,2)
z=f3(6:10)
f3=bitxor(f3(1:5),z)

[p,c3] = divc(f3,y);
```

### inversion.m

```
[q,r2] = division(f2,a2);

u=zeros(1,m+1)
v=zeros(1,m+1);
v(m+1)=1;
v2=conv(v,q);
u(numel(v2)) = 0;
x=bitxor(u,v2);

[q,w2]=divw(f2,x);
```

## 5.6 Conclusion

As a conclusion, in this chapter, the implementation of the project have been discussed. The software development environment for the Simulation of ECC over binary field will explain and show how does the system is being setup by using software and hardware used. As for software configuration management discussed about which software and programming language used for this simulation. Implementation status part described about the description and duration of each module of the system. Lastly, the code generation part showed the whole source code of the system using language of MATLAB. On the next chapter, the testing of the project will be discussed.





## CHAPTER 6

### TESTING

#### 6.1 Introduction

The chapter of testing of this project will be discussed including testing task. There are two main part in the testing task which are test plan and test result and analysis. Testing will be done after the system is complete and ready to be tested. The testing is the last part in developing a system to ensure that all modules that being described in chapter implementation are able to function properly and to test whether is there any error occurred when the system is running. This testing also to test whether this system meets the user requirements or not.



## 6.2 Test Plan

This part generally will be discussed about the testing scope and activities. This task includes test environment.

### 6.2.1 Test Environment

In this part the topic that will be discussed is the environment that will be tested including hardware and software for the system of the Simulation of ECC over binary field. Table 6.2 is the details about the system testing environment.

**Table 6.1 : Test Environment Specification**

System Configuration	Specification (Laptop)
Operating System	Windows 7 Ultimate
Memory (RAM)	2.00 GB
Hard Disk	500 GB
Processor	Intel(R) Celeron(R) CPU 1007U @ 1.50GHz
Software	MATLAB 2014a

### 6.3 Test Result and Analysis

In this part, the system of the Simulation of ECC over binary is being tested whether all the operations which are addition, multiplication and inversion are being calculated or not and whether there is an error while the system is running. Next is the point representation which are point addition and point doubling are being tested. Finally, the last is the computation of point multiplication. This computation involves the two points that are being calculated earlier which are point addition and point doubling.

### 6.3.1 Addition

The Figure 6.1 below shows that the operation of addition. This operation involve exclusive-or between two number which are A and B.

**BINARY FIELD OPERATION**

**Addition**

A = 0 0 0 1 1 0 0 0 0 1

B = 0 0 0 1 1 0 1 0 1 1

C = 0 0 0 0 0 0 1 0 1 0

Figure 6.1 : Result of the Operation of Addition

### 6.3.2 Multiplication

The Figure 6.2 below shows that the operation of multiplication. This operation involve the operation of convolution and division.

**Multiplication**

A = 1 0 1 0 0 0 1

B = 0 0 0 0 0 1 0

f = 0 0 0 0 1 0 0 0 0 0

C = 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 1 0

Figure 6.2 : Result of the Operation of Multiplication

### 6.3.3 Inversion

The Figure 6.3 below shows that the operation of inversion. This operation involve the operation of division.  $r$  is the remainder and  $w$  is the last answer.

The screenshot shows a web application interface for a binary inversion operation. It has a green header with the title 'Inversion'. Below the header, there are four rows of input and output fields, each with a 'Generate' button. The first row is labeled 'A =' and contains the binary sequence '1 1 1 0 0 1 1 1'. The second row is labeled 'f =' and contains '1 1 0 0 1 0 0 1 1 1'. The third row is labeled 'r =' and contains '0 0 0 1 0 1 1 1 0 0'. The fourth row is labeled 'w =' and contains '1 1 0 0 1 0 0 0 1 0'. At the bottom right of the interface is a 'Calculate' button. A large watermark for 'UTEM' is visible in the background.

Figure 6.3 : Result of the Operation if Inversion

### 6.3.4 Point Addition

The Figure 6.4 below shows that the point representation which is point addition. In this computation, operation that involve are the operation of addition, multiplication and inversion. Point P is  $(x_4, y_4)$  and point Q is  $(x_5, y_5)$ . Point P will be add to the point Q. This means  $P+Q = (x_9, y_9)$ .



Figure 6.4 : Result of the Point Addition

### 6.3.5 Point Doubling

The Figure 6.5 below shows that the point representation which is point doubling. In this computation, operation that involve are the operation of addition, multiplication and inversion. Point P is taken from previous that used by point addition. The result from this computation is 2P.

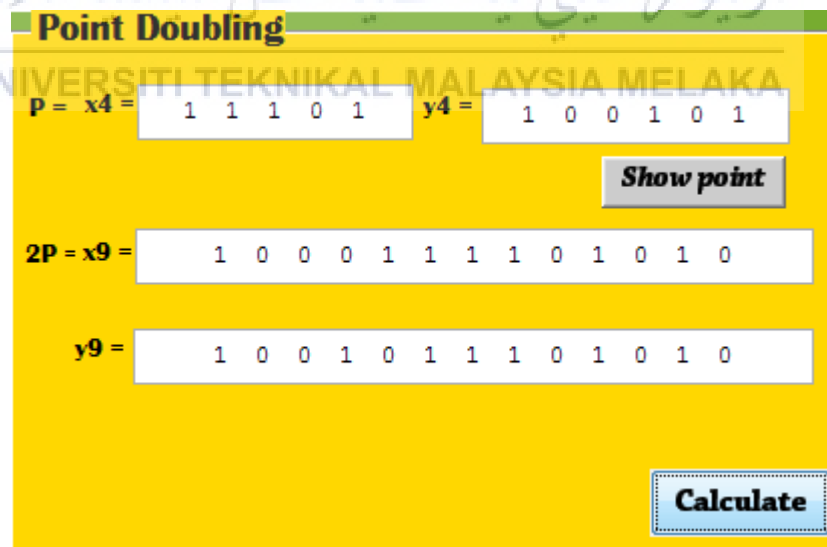


Figure 6.5 : Result of the Point Doubling

### 6.3.6 Point Multiplication

The Figure 6.6 below shows the result of point multiplication. In this computation, the point representation that involve are point addition and point doubling.  $k$  is generated randomly to get the answer of  $Q$ .

**Point Multiplication**  $Q = kP$

$k =$   Generate

$P = x =$

$y =$

$Q = x =$

$y =$

Calculate

Figure 6.6 : Result of the Point Multiplication

### 6.4 Conclusion

In a nutshell, this chapter discussed about the testing of this project. The test plan include the test environment. In the task of test the result and analysis, it shows that all the test result from the Simulation of ECC over binary field and will show whether the test is success or fail. For the last chapter, which is project conclusion, will be discussed about the project summarization, contribution, limitation and future works of the project.

## CHAPTER 7

### CONCLUSION

#### 7.1 Introduction

In this chapter, the conclusion of the whole project will be explained. Project conclusion is the last chapter for this project. In this chapter, will be discussed about the project summarization, project contribution, project limitation, future works and lastly the conclusion of this project. Project summarization is the summarization of the project by stating what are the objectives of the project, the weakness and strength of the project and the whole project criteria which including chapter 1 to chapter 6. Project contribution will be discussed in details according to the previous chapter which is chapter 1 in the subtopic of project contribution. For project limitation, it will state the limit of the project. Finally, for the future works will be discussed on how to improve the system in future.

#### 7.2 Project Summarization

Currently, the process of teaching and learning of Elliptic Curve Cryptography in academic is still needed to calculate by manual different with in industry that uses everything in computerized system. Therefore, to help the teaching and learning process, the system of Simulation of ECC over binary field is develop to enhance the better understanding and computation.

In Elliptic Curve Cryptography, there are two underlying field which are prime field and binary field. Prime field is the operations that are done by modulo prime number while binary field is the operation that are dome by modulo an irreducible polynomial. This project focus on binary field only. Elliptic Curve have

two point representation which are point addition and point doubling. Elliptic Curve also have three basic operations which are addition, multiplication and inversion.

The first chapter would discussed about the problem statement of the project, objectives of the project, project scope and project contribution. Problem statement of this project are because the numbers that will be calculated usually limited to small numbers. Other than that, the academic computation on ECC over binary field before this is still calculated manually and managed by user. Therefore, the objectives of the project is needed to overcome the problem that have been stated. The objectives of this project is to generate random numbers during calculation. Other than that, the objectives of this project is to build and develop a system to calculate cryptosystem of ECC over binary field for user. Lastly, the objective of this project is to enable the user to calculate or compute large numbers on ECC over binary field.

The second chapter would explained about the literature review which we will refer to the books, website, article and journal as reference to study and understand about what is the project in detail. The related terms, related works and critical review of current problem and justification have been explained in this chapter. For the third chapter, the methodology used have been discussed. The methodology used for this project is the V-Shaped Model because it can approved what we have been done in this project and it is the advanced of Waterfall Model.

The next chapter which is chapter four, the design is covered in this chapter. The interface and the logical design which is the flowchart have been explained in this chapter. Chapter five is the implementation of this project. It covered the design development, software installation and configuration, implementation status of the system and the code of the system. Lastly, in chapter 6, the testing have been done to test whether the system run and meet the user requirement or not. The testing has been done for all operation and point representation involve in this project. The last chapter which is in this chapter will cover project contribution, project limitation and future works for the project.



### **7.3 Project Contribution**

This project will contribute to the user to teach and learn how the operation which are addition, multiplication and inversion and the point representation of point addition, point doubling and point multiplication are work by this simulation of ECC over binary field. By developing the simulation, the objectives of this project will be achieved as it have been proven by conducting the testing in chapter six.

### **7.4 Project Limitation**

The limitation of this project is the MATLAB cannot deal with the large integers that we face in cryptography. These problem sometimes can be overcome through the use of alternative algorithms such as those that used for the modular exponentials. However, sometimes these method also not possible to overcome these problems. Even basic operations such as the operation of squaring make the numbers that are too big for MATLAB to store properly. The basic limitation is the way MATLAB stores the result's number. Double precision IEEE floating point used by MATLAB can normally store integers up to 4,503,599,627,370,495 (52 bits). Any number that exceeds this limit will have some less significant digits and thus will loosen the accuracy. The future versions of MATLAB that incorporate the proposed new IEEE standards may be able to accurately store an integer with a bit length roughly double the current maximum, but although it still cannot properly address the 160 bits required to maintain security.

### **7.5 Future Works**

By 2000, there is a tendency that the conventional public key cryptography system, especially the RSA-based system, gradually replaced by ECC system. However, today other cryptographic NTRU contender makes its way into the mainstream market. Another programming language such as Java and Maple also can be used to perform computation and better design of the interface.

## 7.6 Conclusion

In a nutshell, the objectives of the project of the Simulation of ECC over binary field have been achieved successfully through the explanation and discussion for the previous chapter. The system of the Simulation of ECC over binary field have been successfully developed and already tested in chapter six. This system will help the the process of teaching and learning that will be used by lecturer and student.



## REFERENCES

Berta, IstvanZsolt, & Mann, Zoltan Adam (2003). Implementing elliptic curve cryptography on PC and smart card. *Periodica Polytechnica, Electrical Engineering*, 46(1-2), 47-73.

Mishra, A. R. (2012). Elliptic Curve Cryptography ( ECC ) for Security in wireless Sensor Network, *I(3)*, 1-6.

Johnson, D., Menezes, A., & Vanstone, S. (2001). The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security*, 1(1), 36-63.

Barsagade, M. W., & Meshram, S. (2014). Overview of History of Elliptic Curves and its use in cryptography, *5(4)*, 467-471.

Kaur, A., Goyal, V., & Luthra, P. (2013). Java Implementation And Arithmetic Performance evaluation of Elliptic Curve Cryptography Using MATLAB, *2(6)*, 2695-2699.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Silverwood, H. (2007). Summer Research Project A MATLAB Implementation of Elliptic Curve Cryptography Cryptography.

Wang, H., Sheng, B., & Li, Q. (2006). Elliptic curve cryptography-based access control in sensor networks. *International Journal of Security and Networks*, 1(3/4), 127. <http://doi.org/10.1504/IJSN.2006.011772>

Hankerson, D. (2004). *Guide to Elliptic Curve Cryptography*. <http://doi.org/10.1007/b97644>

Tutorial, A. I. (n.d.). Elliptic Curve Cryptography. *Cryptographic Algorithms on Reconfigurable Hardware*, 291-328. [http://doi.org/10.1007/978-0-387-36682-1\\_10](http://doi.org/10.1007/978-0-387-36682-1_10)

Shankar, T. N., & Sahoo, G. (2010). Cryptography By Karatsuba Multiplier with ASCII Codes\_MATLAB. *International Journal of Computer Applications*, 1(12), 54–62. <http://doi.org/10.5120/262-421>

Liu, F. (n.d.). A Tutorial on Elliptic Curve Cryptography ( ECC ).

Guide, A. I. (n.d.). Elliptic Curve Cryptography, 1–11.

Neustadter, D., & Denis, T. S. (2008). Cryptography Elliptic Curve Cryptography ( ECC ).

Neustadter, D., & Denis, T. S. (n.d.). Elliptic Curves over Prime and Binary Fields in Cryptography. Retrieved from [https://www.fields.utoronto.ca/programs/scientific/07-08/cryptography/dana\\_neustadter.pdf](https://www.fields.utoronto.ca/programs/scientific/07-08/cryptography/dana_neustadter.pdf)

Singh, P. K., & Choudhary, M. K. (2013). Scalar Multiplication Algorithms of Elliptic Curve Cryptography over  $GF(2^m)$ . *International Journal of Innovative Technology and Exploring Engineering*, (31), 2278–3075. Retrieved from [https://pdfs.semanticscholar.org/6605/cff0a8404b713af7bfae164745cf769c1e90.pdf?\\_ga=2.33085502.1673830604.1503130303-210559058.1503130303](https://pdfs.semanticscholar.org/6605/cff0a8404b713af7bfae164745cf769c1e90.pdf?_ga=2.33085502.1673830604.1503130303-210559058.1503130303)

Tecnology, I., & Modares, H. (2009). A Scalar Multiplication in Elliptic Curve Cryptography with Binary Polynomial Operations in Galois Field, (October).

Technology, I. (2002). IMPLEMENTING ELLIPTIC CURVE CRYPTOGRAPHY ON PC, 46(1), 47–73.

## BIBLIOGRAPHY

<https://books.google.com.my/books?id=MhvcBQAAQBAJ&printsec=frontcover&dq=Alfred+J.+Menezes,+Paul+C.+van+Oorschot,+Scott+A.+Vanstone,+1996&hl=en&sa=X&ved=0ahUKEwjswvnBmoHUAhUCnJQKHeFrBL4Q6AEIjAA#v=onepage&q=Alfred%20J.%20Menezes%20C.%20Paul%20C.%20van%20Oorschot%20S cott%20A.%20Vanstone%20C%201996&f=false>

<http://scialert.net/fulltext/?doi=itj.2006.204.229>

<http://www.embedded.com/design/safety-and-security/4396040/An-Introduction-to-Elliptic-Curve-Cryptography>

<https://www.certicom.com/content/certicom/en/ecc-tutorial.html>

### **Cryptographic Algorithms on Reconfigurable Hardware**

By Francisco Rodriguez-Henriquez, N.A. Saqib, Arturo Díaz Pérez, Cetin Kaya Koc

UNIVERSITI TEKNIKAL MALAYSIA MELAKA