Design of ECC-based Efficient Arithmetic Operations and Signature Scheme

THESIS SUBMITTED IN PARTIAL FULFILLMEMT OF THE

REQUIREMENT FOR THE DEGREE OF

MASTER OF ELECTRONICS AND TELECOMUNICATION ENGINEERING

BY

NANHE KUMAR SINGH

REGISTRATION NO: 140690 of 2017-18
EXAMINATIONS ROLL NO: M4ETC19001

SUPERVISOR:

DR. JAYDEB BHAUMIK

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

JADAVPUR UNIVERSITY, KOLKATA-700032 INDIA

2019

DECLARATION

I hereby submit this thesis for partial fulfillment of MASTER OF ENGINEERING IN ELECTRONICS AND TELECOMMUNICATION ENGINEERING degree. I also declare that this thesis contains original work and related literature survey. All information in this document have been obtained and presented in accordance with academic rules and ethical conduct. I also declare that as required by these rules and conduct I have fully cited and referenced all materials and results that are not original with this work.

NAME: NANHE KUMAR SINGH

REGISTRATION NO: 140690 of 2017-18

EXAMINATION ROLL NO: M4ETC19001

PROJECT TITLE:

Design of ECC-based Efficient Arithmetic Operations and Signature Scheme

DATE:

PLACE:

____

(NANHE KUMAR SINGH)

Faculty of Engineering & Technology

Jadavpur University

CERTIFICATE OF RECOMMENDATION

This is to certify that the thesis entitled **“Design of ECC-based Efficient Arithmetic Operations and Signature Scheme”** has been carried out by **Nanhe Kumar Singh** (Examination Roll No: M4ETC19001 and Registration No:140690 of 2017-18) under my guidance and supervision and be accepted in partial fulfillment of the requirement for the degree of Master of Engineering in Electronics & Telecommunication Engineering.

Countersigned by:

Dr. Jaydeb Bhaumik
Thesis Supervisor
Dept. of ETCE
Jadavpur University
Kolkata-700032
West Bengal, India

Prof. Dr. Sheli Sinha Chaudhuri
Head, Dept. of ETCE,
Jadavpur University,
Kolkata 700032
West Bengal, India

Prof. Chiranjib Bhattacharjee
Dean, FET
Jadavpur University, Kolkata
West Bengal, India

Faculty of Engineering & Technology

Jadavpur University

CERTIFICATE OF APPROVAL*

The forgoing thesis, entitled **“Design of ECC-based Efficient Arithmetic Operations and Signature Scheme”** is hereby approved as a creditable study on an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

Committee on Final Examination for
Evaluation of the Thesis:

External Examiner

Dr. Jaydeb Bhaumik
SUPERVISOR

*Only in case the thesis is approved.

ACKNOWLEDGEMENT

This is a great opportunity to express my gratitude and respect to all the persons who have supported me, each, in his/her own way, during my Master course.

It is difficult to overstate my gratitude to my respected Supervisor, Dr.Jaydeb Bhaumik, for guiding and inspiring me throughout my thesis period at Jadavpur University. I highly appreciate him for making me understand what research is. Moreover, I will always remain indebted to him for encouraging and giving me so much time, in spite of his busy schedule.

I would like to thank Prof. Dr. Sheli Sinha Chaudhuri, H.O.D, Dept. of ETCE, Jadavpur University for providing me all the facilities for carrying out the entire project work.

I would also like to thank our Industrial Lab members for their support and inputs during my research.

May 2019

Abstract

In recent years, the internet of things (IoT) has beena focus of research. Elliptic curves can be used in Light weight IoT authentication protocol. Elliptic curves are used for several kinds of cryptosystems, even it involved in key exchange protocols and digital signature algorithms [11], since it independently presented by Miller [20] and Koblitz [15] in the 1980s.it requires a shorter key length compared to other public-key cryptosystems such as RSA.A shorter key reduces the required computations, power consumption, and storage.The major time-consuming operation in Elliptic Curve Cryptography (ECC) is the point multiplication, kP. Therefore, a lot of research has been carried out to improve the efficiency of ECC implementations. Composite Elliptic Curve (EC) operations and recoding methods are two factors that affect the efficiency of EC scalar multiplication.Deciding which composite EC operation to be used in an ECC system helps to improve the computational efficiency.

In this thesis, we focus in optimizing such ECC operation at the point and scalararithmetic levels, specifically targeting standard ECC curves over prime fields. Therefore, comprehensive information related to EC multiplication methods will be provided in order to facilitate literature review for researchers who would like to conduct a research in this area of science. In this a research, a survey of EC single scalar multiplication methods is introduced. At the scalar arithmetic level, we develop new sub linear (in terms of Hamming weight) multibase scalar multiplications based on Non Adjacent Form(NAF) like conversion algorithms that are shown to be faster than any previous scalar multiplication method. We verified the result in python code and also proposed scalar multiplier is embedded in ECDSA. After that we can conclude that signature generation and verification is faster than existing method.

We also focus on a fast point doubling and point addition operations on an elliptic curve over prime field .This occur when we use a special coordinates system to represent any point on elliptic curve over prime field. Using this mixed co-ordinate system the computation cost for point doubling operation and point addition operation are reduced.

Elliptic curve cryptography (ECC) is extensively applied in various multifactor authentication Protocols. In this work, a recent ECC based authentication protocol isanalyzed using threats model and static analysis to detect the vulnerabilities, and to enhance its security against existing threats. If protocols are vulnerable,damages could include critical data loss and elevatedprivacy concerns.The protocols considered in thiswork differ in their usage of security factors encryption and timestamps.The threat model considers different types of attacks includingman in the middle, weak authentication and denial of service. Countermeasures to reduce or prevent such attacks are suggested.

Table of Contents

Abstract

Acknowledgements

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Motivation………………………………………………………………………. 1

1.2 Significance of this work……………………………………………………….. 3

1.3 Thesis Outline……………………………………………………………………….4

Chapter 2: Background

2.1 Preliminaries……………………………………………………………………..5

2.2 Cryptography and Its Components…………………………………………………6

2.3 Symmetric Key and Asymmetric Key……………………………………………..7

2.4 Hash function And Uses of Hashing………………………………………………9

2.5 Introduction of Elliptic Curves…………………………………………………….9

2.6 Mathematics of Elliptic Curves Over FieldFp…………………………………….11

2.6.1 Point Addition and doubling……………………………………………………….13

2.6.2 Point Multiplication…………………………………………………………...18

2.7 ECC Curves Parameter ………………………………………………………………..19

2.7.1 Domain Parameter of ECC over Prime Field……………………………………..19

2.7.2 Domain Parameter of ECC over Binary Field…………………………………….19

2.8 ECDH – Elliptic curve Diffie Hellman…………………………………………………...19

2.9 NIST Recommended ECC curves For Cryptography……………………………….20

2.10 Elliptic Curves Discrete Logarithmic Problem……………………………………..24

2.11ECC Attacks…………………………………………………………………………24

2.12 ECC attacks Countermeasures ………………………………………………..…….26

Chapter 3 : Analysis of Attacks in ECC based Protocol

3.1 Introduction………………………………………………………………………. 27

3.2 Application of the Threat Model to Analyze Protocols……………………………...28

3.3General Algorithm and Condition for Various Types of Attack………………………. 28

3.3.1 Clogging Attack…………………………………………………………...28

3.3.2 Data Base Attack……………………………………………………… 31

3.3.3 Man In Middle Attack………………………………………………… 32

3.3 Basics of RFID……………………………………………………………. 33

Chapter4:Our Model

4.1 Model Description …………………………………………………......... 34

4.2 Flow Chart of our Model……………………………………………………..35

4.3 Elliptic Curve Digital Signature Algorithm……………………………………..37

4.4 Accelerating ECC Operation……………………………………………………40

4.5 Final Algorithm For scalar Multiplication ……………………………………...42

4.6Moosavi et al.'s Protocol for RFID Implant Systems…………………………………42

4.6.1Clogging Attack onMoosavi et al.'s Protocol………………………………………45

4.7 Proposed Countermeasures from the Attack……………………………………... 45

Chapter 5: Implementation Results

5.1 ECC Addition Execution Time ……………………………………………………47

5.2 ECC Doubling Execution Time……………………………………………………..49

5.3 ECC Scalar Multiplication Execution Time ………………………………………..50

5.4 Signature Generation Time ……………………………………………………52

5.5 ECDSA Execution Time...52

5.6 Summary……………………………………………………………………………..52

Chapter 6: Conclusion and Suggestion for Future Work

6.1 Conclusion …………………………………………………………......54

6.2 Future Scope……………………………………………………...... 54

6.3 References………………………………………………….........55

List of Tables:

Table 2.1 Comparison between ECC and RSA key size……………………….8
Table 2.2 Point Conversion Complexity………………………………………..16
Table 2.3 Comparison of Computational Cost for Doubling…………………...17
Table 2.4 Bit Lengths of the Underlying Fields of the Recommended Curves…21
Table 2.5 NIST Recommended Elliptic Curves parameters…………………....23
Table 2.6 Summary of ECDLP attacks and their countermeasures…………….26

Table 3.1 RFID-Frequencies……………………………………………………..35

Table 4.1 Computational Cost of ECC addition in different Co-ordinates………..42

Table 5.1 Execution time of ECC addition……………………………………..48
Table 5.2 Execution time of ECC doubling operation…………………………….51
Table 5.3 Execution Time of ECC scalar Multiplication…………………………51
Table 5.4 Execution Time of ECC based Signature generation Scheme…………53
Table 5.5 Execution Time of ECDSA scheme……………………………………53

List of Figures:

2.1 Block diagram of two party communications 6
2.2 Block Diagram of Symmetric Key Encryption and Decryption 7
2.3 Asymmetric Key Based Encryption... 8
2.4 Elliptic curve Mathematical Hierarchy...11
2.5 ECC Point Addition...13
2.6 ECC Point Doubling...14

3.1 Threat Model... 27
3.2 Threat Model Analysis Steps.. 28
3.3 Clogging Attack.. 29
3.4 Comparison of ECC and Modex operation.. 30
3.5 Data Base Attack.. 31
3.6 Man in middle Attack... 32

4.1 Flow chart of proposed work.. 36
4.2 Block Diagram of Signature Generation and Verification Process......................... 37
4.3 ECDSA Block Diagram... 38

5.1 ECC Addition: Memory vs Time graph in Affine co-ordinates for NIST192$_P$...........47
5.2 ECC addition: Memory vs Time graph in Chudnovsky co-ordinates for NIST192$_P$...48
5.3 Time vs Memory graph for doubling in Affine co-ordinates for NIST192$_P$...............50
5.4 Time vs Memory graph for doubling in modified Jacobian co-ordinates for NIST192$_P$...51
5.5 Time vs Memory graph of scalar multiplication in affine co-ordinate NIST192p.....52
5.6 Time vs Memory graph of scalar multiplication for NIST192p.............................. 52

Chapter 1

Introduction

1.1 Motivation

Since a lot of sensitive data such as credit card numbers and social security numbers are transmitted over the Internet during transactions. Securing electronic transaction becomes a very important issue. An efficient way to protect and secure the information is by using cryptography which can be used to provide and assure confidentiality and integrity of the transactions (Mackenzie, et al. 1996).

The history of cryptography is long and interesting. It had a very considerable turning point when two researchers from Stanford, Whitfield Diffie and Martin Hellman, published the paper “New Directions in Cryptography” in 1976. They preface the new idea of public key cryptography in the paper.

Public-key cryptography and symmetric-key cryptography are two main categories of cryptography. The Well-known public-key cryptography algorithms are RSA (Rivest, et al. 1978), El-Gamal and Elliptic Curve Cryptography.

Elliptic curve cryptography (ECC) was independently introduced by Koblitz and Miller in 1985. Since then, this public-key cryptosystem has attracted increasing attention due to its shorter key size requirement in comparison with other established systems such as RSA and Discrete Logarithm (DL)based cryptosystems. For instance, it is widely accepted that 160-bit ECC offers equivalent security as 1024-bit RSA. This significant difference makes ECC especially attractive for applications on constrained environments as shorter key sizes are translated to less power and storage requirements, and reduced computing times.

Scalar Multiplication is Denoted by dP, where d is the secret key (scalar) and P a point on the elliptic curve, the scalar multiplication is the central operation of elliptic curve cryptosystems. The computation of this operation involves three mathematical levels: field arithmetic, point arithmetic and scalar arithmetic. Significant effort to optimize ECC operations through each of those levels has been carried out through the last few years. In this work, we concentrate efforts at the point and scalar arithmetic levels, specifically for the case of standard curves [NIST] over prime fields.

ECC point arithmetic involves the efficient execution of doubling and addition operations. At this level, several authors have been working on making those basic point operations as efficient as possible. Although the idea of combining basic operations to build more sophisticated point operations was around years ago, it mainly focused onaffine coordinate's formulae, which are particularly inefficient because they contain expensive field inversions.

An elliptic curve can be represented using several coordinates system. For each such system, the speed of point doubling and point addition operations is different. This means a good choice ofcoordinates system is an important factor for speeding up the elliptic curve scalar multiplication.

In the scalar arithmetic level, efforts have mainly focused on developing efficient numeric expansions for integers (scalar d in our case) that reduce to the minimum the number of point operations required for the computation of the scalar multiplication. The well-known NAF is traditional and efficient examples for the latter.Progress in the point arithmetic level with the introduction of the mixed co-ordinates mentioned previously has benefited this area with the appearance of the fastest scalar multiplication known in the literature.

Besides this work we additionally analyzed Security analysis of a ECC based Protocol for RFID. We analyzed different types of attack and also mount an attack in the   protocol. Then we provide a Counter measure of the attacks.

1.2 Significance of this work

Our work focuses on the optimization of the ECC scalar multiplication at point and scalar arithmetic levels for the case of standard curves over prime fields the proposed scalar multiplications, which can be applied to ECC over prime field.

The contributions of thesis are as follows:

1. Accelerating the traditional scalar multiplication by exploring the NAF algorithm with mixed co-ordinates. we achieved efficient computational time compared to traditional one.
2. Selection of appropriate co-ordinates system and appropriate ECC Curves accordance with system requirements.
3. Improve Computational cost in ECC addition and doubling operation while using inversion free best co-ordinates that are suited for operation.
4. Embeddingof proposed scalar multiplier in Elliptic curve Digital Signature Algorithm (ECDSA) makes generation of public key, signature generation and signature verification efficient than previous one.

5. The other goal of this work is to perform static analysis on the underlying vulnerabilities and security threats that exist in ECC based protocol that are implemented in RFIDs. And also design possible countermeasures to defeat the identified vulnerabilities in this protocol.

1.3 Thesis Outline

The organization of the work is as follows.

Chapter 2 provides the basic of elliptic curves and literature reviews also it provide the   reader to introduction of each arithmetic level that constitutes the scalar multiplication.

In chapter 3 we analyze the security threat in ECC based protocol. And also Basics of RFID have been discussed in this chapter which is necessary to analyze RFID based protocol.

In chapter 4 we describe our methodology by flow chart. And we proposed efficient doubling and addition operation for further improvement of computational cost in scalar multiplication. The proposed Scalar multiplication is embedded in ECDSA and it is shown that the signature generation is more efficient than traditional one. We analyzed moosavi et.al protocol and describe the attacks carried out on protocol and suggesting the countermeasure of the attacks.

Chapter 5 presents the implementation and results. In this chapter, we compared our result with existing one. Plots of memory vs time for different ECC based arithmetic operations are also presented in this chapter.

In chapter 6, conclusions and suggestions for future work are presented.

Chapter 2

2.1 Preliminaries:

First we are introducing some concept of group and Field before ECC Introduction.

Groups:

A **group** *G*, sometimes denoted by {G , .} is a set of elements with a binary operation, denoted by ·, that associates to each ordered pair (a ,b) of elements in G an elements(a.b) in G such that following Axioms are obeyed.

Closure: Ifaandb∈ *G*, then $a \cdot b$ is also ∈*G*.

Associative: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ for all *a*, *b*, *c*∈ *G*.

Identity element: There is an element e in *G* such that $a \cdot e = e \cdot a = a$ for all *a* ∈ G.

Inverse element: For each *a* in *G* there is an element *a*' in *G* such that a.a' = a'.a = e

If a group has a finite number of elements, it is referred to as a **finite group**, and the **order** of the group is equal to the number of elements in the group. Otherwise, the group is an **infinite group**.

Field: Afield F, sometimes denoted by {F, +, x}, is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c ∈F the following axioms are obeyed:F is an integral domain; that is, F satisfies Closure, Associative, Identity element, Inverse element,Closure under multiplication,Associativity of multiplication, Distributive laws,Commutativity of multiplication, Multiplicative identity, No zero divisorsandMultiplicative Inverse.If a, b ∈Fand ab= 0, then either a = 0 or b = 0.For each a ∈ F, except 0, there is an element a^{-1}∈ F such that $a^{-1}.a = a.a^{-1} = 1$

Finite Fields:

We will explain the concept of finite fields through the definition of the finite field F*p*that is used through this work.As an extension of the definition of groups, the prime field F*p*is finite since it contains afinite number of elements given by *p*, which also represents the order of the field. In a more general sense, given a field Fqof order *q*, it is said to be a finite field if and only if its order is a prime power $q = p^m$.

2.2Cryptography and its Components

Cryptography

It enables people to store or transmit sensitive information via insecure network.Cryptography uses mathematics to designing of secret codes. On the other hand, cryptanalysis is the science of breaking of secret codes. Employing encryption and decryption, two persons, Alice and Bob, can communicate via an insecure channel in a secure way. The third person who is eavesdropper (Eve, abbreviated as E) should not be able to read the clear text or change it.

The goal of cryptography is to exchange messages using cryptography which are not understood by other people. Figure 2.1 provides a basic model of a two-party communication using encryption. In this model, an entity is a person or a device that sends, receives or manipulates data.Senderis an entity that legitimately transmits the information. On the other hand, a receiveris an entity that is the recipient of information.An adversary plays the role either as the sender or the receiver. The other synonymous names for adversary are attacker, enemy, eavesdropper, opponent and intruder [15].

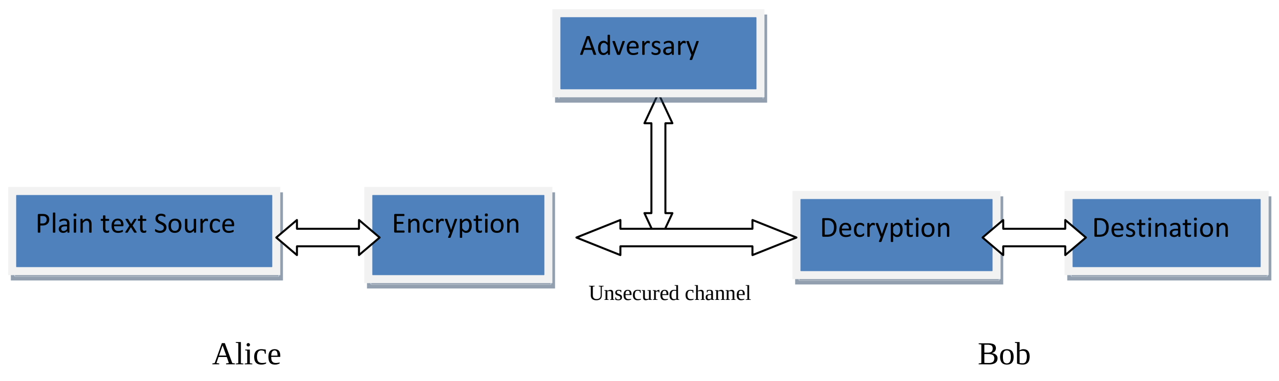


Figure 2.1: Two Party Communications

Cryptography Components

In order to secure messages, there are mathematical techniques that provide security services such as confidentiality, integrity, authentication and non-repudiation.

Confidentiality: Theterm confidentiality means kept the data privately in communication.It can be achieved by encryption .It protects the transmitting data between two entities. It similarly secures the traffic flow analysis.

Integrity: Data is not changed in un-authorised manner.It is typically provided by digital signature and encryption as well.

Authentication: Receiver determines its source to confirm the sender's identity by using something that you have or you know. Normally, it is done by using the sender public key. It is the same integrity provided by digital signature.

Non-Repudiation:It ensures the sender and receiver from denying the sending or receiving of a message and the authenticity of their signature. Typically, it is provided by digital signature [14].

2.3 Symmetric Key and Asymmetric Key

Symmetric Key

In Symmetric key same key is used for encryption as well as decryption. In Symmetric Key cryptography each party must trust each other and not tell the secret-key to anyone else. The efficient encryption of large amount of data is the advantage of the symmetric-key, however, the problem appears when key management is over the large number of user needs,[1, 2].Figure 2.2 shows the block diagram of symmetric key encryption and decryption.

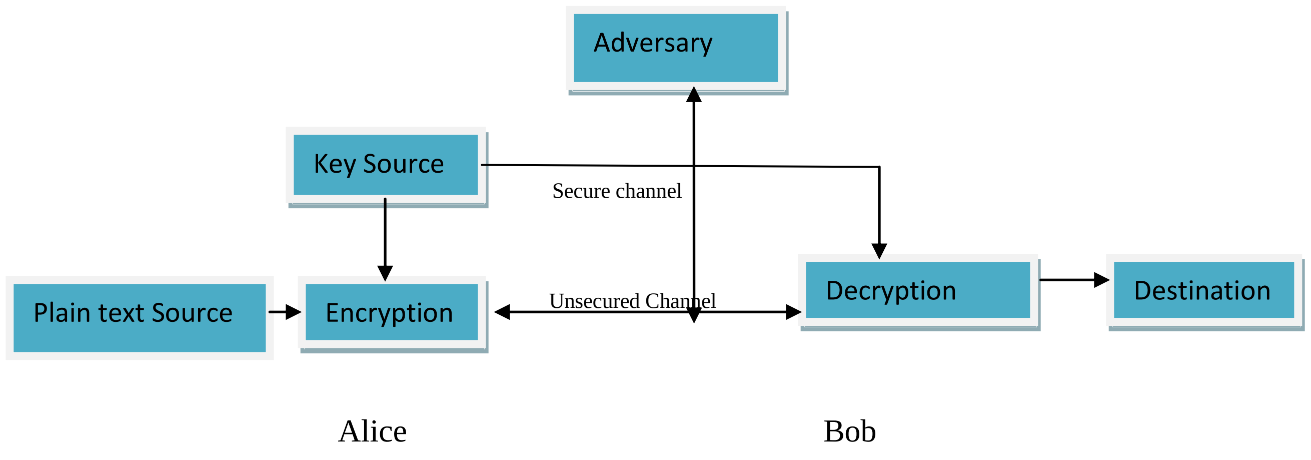


Figure 2.2 Block Diagram of Symmetric Key Encryption and Decryption

Symmetric Key Advantages

1. Symmetric-key ciphers can be considered in order to have high rates of dataThroughput. Some hardware implementations get hundreds of megabytes per second for encryption rate. Software implementations get megabytes per second in the range

2. Symmetric-key ciphers contain keys that are relatively short.

Disadvantages of Symmetric Key

1. The key must remain secret at both ends in communication.
2. Key management is one of the big problems in large networks.
3. Symmetric-key encryption typically requires a large key for the public verification Function in Digital signature mechanisms [16].

Asymmetric Key (Public Key)

Diffie and Hellman first publicly introduced the concepts of public-key cryptography in 1976.Public key cryptography contains two keys, which are public and private keys. Asymmetric algorithms rely on one key for encryption and a different but related key for decryption. A Situation is assumed where Alice wants to send a message to Bob. Alice uses Bob's Public key to encrypt a message and her private key to sign the message. Bob (receiver) uses his Private Key to decrypt the message and he uses Alice's Public Key to verify the Signature. The standard bodies have set the key size of the encryption key, in order to provide the desired security.A scenario of a publickey encryption scheme is depicted in Figure 2.3.

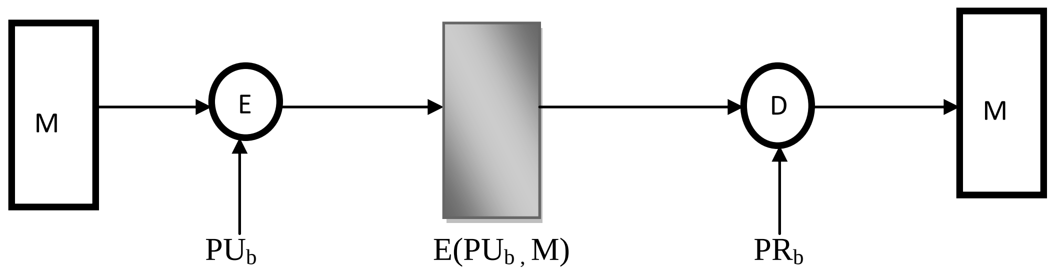


Fig 2.3 Asymmetric Key Based Encryption

Asymmetric Key Advantages

1. In public key cryptography only private key must be kept secret.
2. The number of keys needed in public-key may be significantly smaller than the symmetric-key in a large network.

3. Relatively, efficient digital signature mechanisms can be obtained by many public-key schemes.

Asymmetric key Disadvantages

1. Data throughput rates of most popular public-key encryption methods are several orders slower than the best-known symmetric-key schemes.
2. Key sizes are much bigger than those used in symmetric-key encryption.

2.4 Hash Function and Its Uses

A hash value h is generated by a function H of the form

$$h = H(M)$$

Where M is a variable length message and H(M) is a fixed length values.

The hash value is appended to the message at the source at a time when the message is assumed or known to be correct. The receiver authenticates that message by recomputing the hash value.

The purpose of a hash function is to produce a "fingerprint" of a file, message, or other block of data. The main idea of hashing is to provide confidence and security of the authentic data transmission to user.A hash function H must have the following properties

1. H can be applied to a block of data of any size.
2. H produces a fixed-length output.
3. For any given value h, it is computationally infeasible to find x such that $H(x) = h$. it is calledone-way property.
4. It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.

2.5 Introduction of Elliptic Curves

Elliptic curve cryptography (ECC) was proposed in 1985 by Neal Koblitz and Victor Miller. Elliptic curve cryptographic schemes can provide the same functionality as RSA schemes which are public-key mechanisms. The security is based on the difficultly of a different problem, which is called the Elliptic Curve Discrete Logarithm Problem (ECDLP).

It is offered by other traditional public key cryptography schemes used nowadays, with smaller key sizes and memory requirements. For example, it is generally accepted that a 1024-bit RSA key provides the same level of security as a 160-bit elliptic curve key. The advantages can be achieved from smaller key sizes including storage, speed and efficient use of power and bandwidth. The use of shorter keys means lower space requirements for key storage and quicker arithmetic operations. These advantages are essential when public-key cryptography is applied in resource constrained devices, such as in mobile devices or RFID. These advantages of ECC are employed in this thesis to design cryptosystems for RFID.

Comparison of Key Sizes With Respect to the Computational Effort for Cryptanalysis

Symmetric Scheme (key size in bits)	ECC-Based Scheme (size of n in bits)	RSA/DSA (Modulus size in bits)
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

Table 2.1: Comparison between ECC and RSA key size[4]

Types of Elliptic Curves:

There are two types of Elliptic curves .first one is Elliptic curves over prime field while other is elliptic field over binary field.

The elliptic Field over Binary field F_2m:

The equation to retrieve all the possible points on the curve over the binary fieldF_2m is$y^2 + xy = x^3 + ax^2 + b,$where $b \neq 0$. The elements of the finite field are integers of length at most *‘m’* bits, which they can be considered as a binary polynomial of degree *m – 1*. On the other hand, in binary polynomial the coefficients can only be 0 or 1. Thus, all the operations such as addition, substation, multiplication and division include polynomials of degree *m – 1* or lesser. The m is the finitely large number of points on the EC to make the cryptosystem secure [5].

The elliptic curves over Prime field F_p:

The equation to retrieve all the possible points on the curve over the prime field is $Y^2 mod\ p = x^2 + \mathrm{ax} + \mathrm{b\ mod\ p}$, where $4\mathrm{a}^3 + 27\mathrm{b}^2 \mathrm{\ mod\ p} \neq 0$. The elements of the finite field are integers between *0* and *p – 1* thus all the operations such as addition, subtraction, multiplication, division include integers between 0 and *p – 1*. The primenumber*'p'* is the finitely large number of points on the EC in order to make the cryptosystem secure [5].

2.6 Mathematics of Elliptic Curves over Prime Field F_p

The mathematical hierarchy of the ECC scalar multiplication consists of threelevels:scalar arithmetic, point arithmetic and field arithmetic.

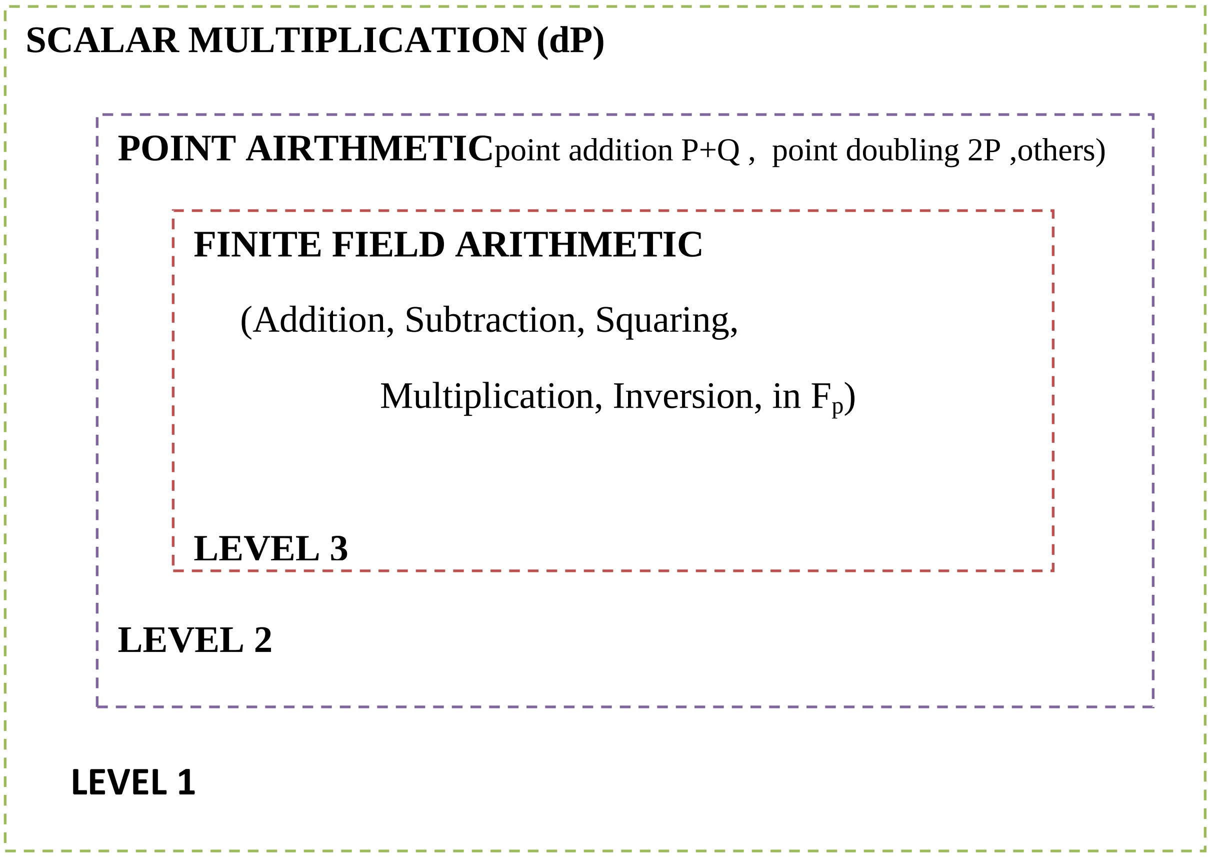

SCALAR MULTIPLICATION (dP)
POINT AIRTHMETICpoint addition P+Q , point doubling 2P ,others)
FINITE FIELD ARITHMETIC
(Addition, Subtraction, Squaring,
Multiplication, Inversion, in F_p)
LEVEL 3
LEVEL 2
LEVEL 1

Fig 2.4 Elliptic curve Mathematical Hierarchy

Finite Field Arithmetic

Basic curve operations in ECC over prime fields are performed using field operations. Thelatter consists of traditional arithmetic operations performed modulo the prime p:

- **Addition:** given a , b ∈ F_p , $a + b \, mod \, p = r$where r is remainder of a+b dividing by p , and $0 \leq r \leq p - 1$.
- **Subtraction:**given a , b ∈ F_P , $(a - b) mod \, p = r$ where r is remainder of $a - b$ divided by p,and $0 \leq r \leq p - 1$.This operation is commonly replaced by an addition performed by a and (-b) ,given that –ve of any element is easily obtained.
- **Multiplication:**given a , b ∈ F_P, $a.b \, mod \, p = r$ where r is remainder of a.b dividing by p , and $0 \leq r \leq p - 1$.
- **Squaring:**given a ∈ F_P, $a^2 \, mod \, p = r$ where r is remainder of a^2 dividing by p,and$0 \leq r \leq p - 1$.
- **Inversion:** given a, is non-zero elements in F_P,$a^{-1} \, mod \, p = r$ is a unique integer in F_p for which $a.p \, mod \, r = 1$.

Example: Given the elements ofF_{11}: { 0 , 1, 2 , ….... ,10} examples of field operations over such finite field are:

- $6 + 8 = 14 = 3 \, mod \, 11$
- $5 - 9 = -4 = 7 \, mod \, 11$
- $7 * 6 = 42 = 9 \, mod \, 11$
- $8^2 = 64 = 9 \, mod \, 11$
- $6^{-1} = 2 \, mod \, 11 \, since \, 6.2 = 12 = 1 \, mod \, 11.$

Level 2 Point Arithmetic:

Scalar multiplication directly depends on operations over points on the elliptic curve *E*. In general, traditional methods to compute the scalar multiplication rely on the execution of a given sequence of point doubling (2*P*) and point addition (*P*+*Q*) operations, where *P* and *Q* are points on the elliptic curve *E*.

Formulae to compute the previous elementary point operations are derived according to what is best known as group law.

Group Law:

To best understand the way point formulae are derived, elementary point operations are typically described geometrically. The following description is based on the naturalrepresentation of points using *x* and *y* coordinates, which is called in the context of ECCaffine coordinates representation.

To addition of two point P and Q in elliptic curve over $F_P$draw a straight line between them and find the third point of intersection *R*. It is easily seen that there is a unique point *R* that is the point of intersection (unless the line is tangent to the curve at either *P* or *Q*, in which case we take *R* = *P* or *R* = *Q*, respectively). We need to define addition on these three points as follows: *P* + *Q* = *R*. That is, we define *P* + *Q* to be the mirror image (with respect to the *x* axis) of the third point of intersection.

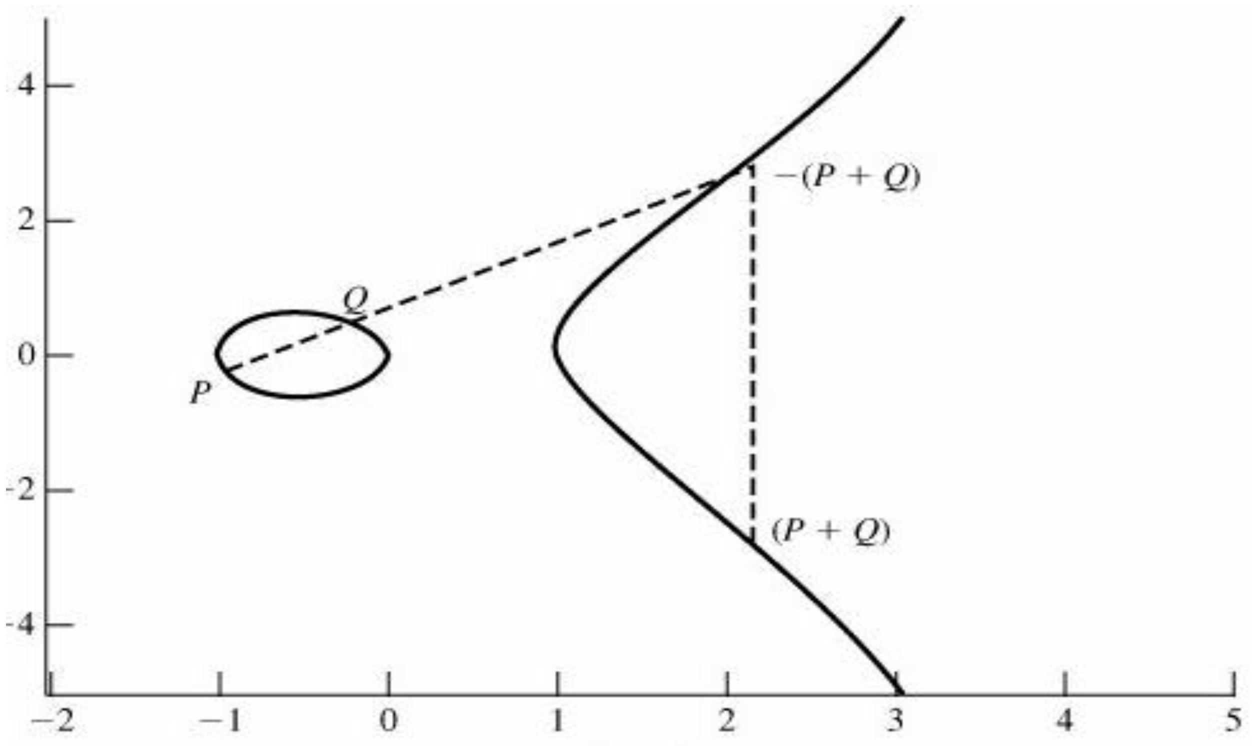


Fig2.5: ECC Point Addition

2.6.1 Algebra for point Addition and Doubling is given below:

In affine co-ordinate let P =(X1, Y1), Q = (X2, Y2) are two points on elliptic curves and the resultant of addition of the two point is a third point R = (X3, Y3).

If P ≠ Q then point addition formula is

$X3 = S^2 - X1 - X2$

And $Y3 = S(X1 - X2) - Y1$

Where $S = \frac{Y2-Y1}{X2-X1}$

The cost of the previous formula is 1I + 2M + 1S.

If P = Q then addition of P+Q = R = 2P .it is doubling of a point.

$X3 = S^2 - 2X1$

$Y3 = S(X1 - X3) - Y1$

Where $S = (3X1^2 + a)/2Y1$

The cost of the previous formula is 1I + 2M + 2S.

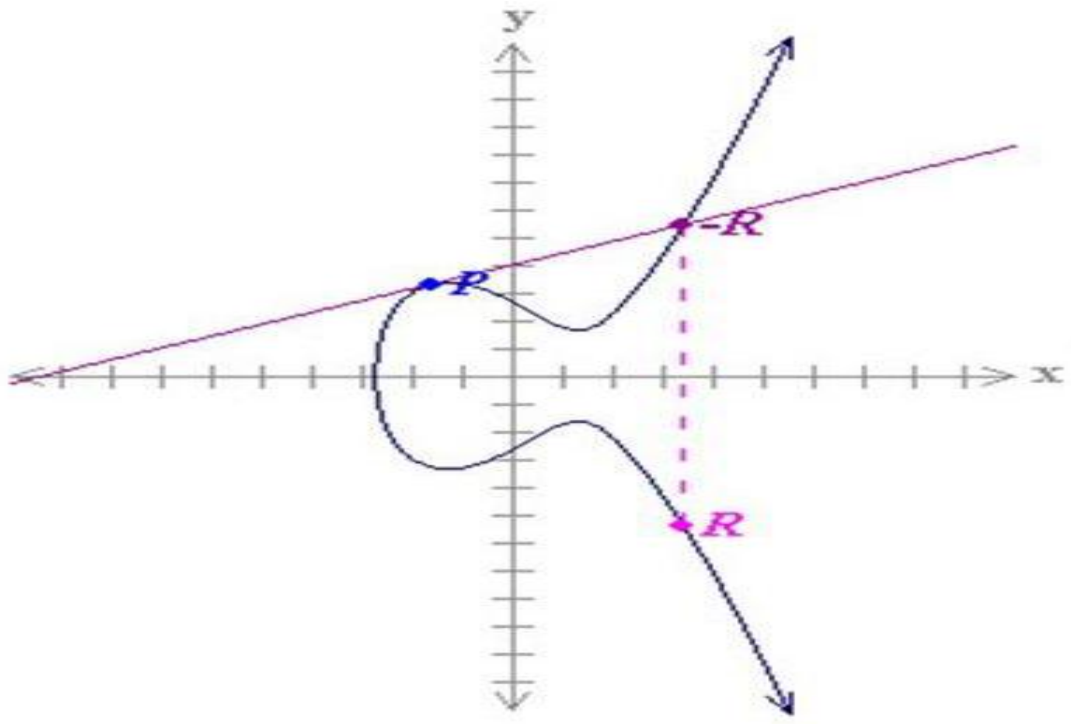


Figure 2.6: ECC Point Doubling

The negative of a point P is the point with the same x co-ordinates but –ve of Y co-ordinates. For example if P =(X, Y) the –P = (X, -Y) and these two point can joined by vertical point so that P+(-P) =O (point at infinity).

Inversion free (Projective co-ordinates)

The representation of points on the elliptic curve E with two coordinates (x,y), namely affine coordinates, introduces field inversions into the computation of point doubling and point addition. Inversions over prime fields are the most expensive field operation and are avoidedas much as possible. It has been observed that, especially in the case of efficient forms for the prime p as recommended by [NIST], $1I > 30M$ [6, 7].

Projective coordinates (X,Y, Z) solve the previous problem by adding the third coordinate Z to replace inversions with a few other field operations. The foundation of these inversion free coordinate systems can be explained by the concept of equivalence class, which is defined in the following.

Given a field K, there is an equivalent relation ≡among non-zero triplets over K, such that [7,8]:

$$(X1, Y1, Z1) \equiv (X2, Y2, Z2) <=> X2 = \lambda^c X1, Y2 = \lambda^d Y1, Z2 = \lambda Z1 \; for \; \lambda \epsilon K \; and \; c, d \; \varepsilon \; Z^+$$

Thus the equivalence class of projective point denoted by (X: Y: Z)is:

$$(X: Y: Z) \equiv (\lambda^c X, \lambda^d Y, \lambda Z) \; where \lambda \in K \; and \; c, d \; \in Z^+ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.1)$$

It is important to remark that any (X, Y, Z) in the equivalence class (2.1) can be used as a representative of a given projective point. In this case, there is only oneelement that can be represented byZ = 0, which is the point at infinity O. The latter can be obtained as follows:

$$(X/_{Z^c}, Y/_{Z^d}, 1)$$

In this thesis, we work with modified Jacobian coordinate, a special case of projective coordinate that has yielded very efficient point doubling andmixed Jacobian co-ordinatefor addition formulae for ECC over prime fields.

Jacobian coordinates representation is obtained by fixing $c = 2$ and $d = 3$ in (2.1). Thus, the equivalence class for Jacobian coordinates is as follows:

$$(X: Y: Z) = (\lambda^2 X, \lambda^3 Y, \lambda Z) \; where \; \lambda \; \varepsilon \; K$$

For converting Jacobian to affine co-ordinate we replace x = X/Z^2, y = X/Z^3

In the specific case of addition, representing one of the points in Jacobian and the other in affine coordinates has yielded the most efficient addition formula, which is known as mixed addition in affine-Jacobian coordinates.

In order to use mixed coordinates we may need to change coordinates. Table 2.2[23] shows the number of field operation required to convert from one set of coordinates to another. In this table, M denotes field multiplication or squaring cost; and I denote field inversion cost for hardware implementation. However, in the remaining table in this section, M and S denotes the cost of field multiplication and field squaringrespectively.

To / From	**Affine**	**Projective**	**Jacobain**	**Chudnovsky**
Affine	--	--	--	--
Projective	2M+I	--	2M+I	2M+I
Jacobian	4M+I	4M+I	--	2M
Chudnovsky	4M+I	4M+I	--	--

Table 2.2 Point Conversion Complexity

It is important to note that in the case of doubling it has been suggested to fix the parameter a =-3 for efficiency purposes. In fact, most curves recommended by NIST for public-key use a = -3, which has been shown to not impose significant restrictions to the cryptosystem [9].

From Table 2.3 [23] shows a comparisonin terms of computationalefficiency fordoubling operation for a arbitrary ‘a’ and for the value of a = -3, where a is the elliptic curve parameter.

Computational cost of EC doubling operation			
Doubling (arbitrary a)		Doubling with a =-3	
2A -> A	2M+2S+I	2A -> A	2M+2S+I
2J -> J	7M+5S	2J -> J	7M+3S
2P -> P	4M+6S	2P -> P	4M+4S
2C -> C	5M+6S	2C -> C	5M+4S

Table 2.3 Comparison of Computational Cost for Doubling, A= Affine, P= Standard projective, J= Jacobian, C= Chudnovsky

GeneralPoint addition in Jacobian co-ordinates

Let P1 = (X1, Y1, Z1) and P2 = (X 2,Y2, Z2) be points in a non singular elliptic curve in projective coordinates on E(Fp) with P1 ≠P2 . The formula for the point addition operation P1 +P2 = (X3,Y3,Z3) is provided in [10]andfor the sake of completeness, expressions are presented as follows.

$X3 = D^2 - E^3 - 2X1Z2^2E^2$

$Y3 = D(3X1Z2^2E^2 - D^2 + E^3) - Y1Z2^3E^3$

$Z3 = Z1^3Z2^3E^3$

With $D = Y2Z1^3 - Y1Z2^3 \; and \; E = X2Z1^2 - X1Z2^2$

For addition no of operations require 12M+4S. If Z1 =Z2 =1 then computational cost is reduces to 3M+3S.

Point Doubling in Jacobian co-ordinates

Let P = (X,Y, Z) be a point in a non singular elliptic curve in Jacobian coordinates on E(Fp) . Then the formula for the point doubling operation 2P = ($X3$,$Y3$, $Z3$) is as follows:

$X3 \; = \; B^2 - 2A$

$Y3 = \; B(A - X3) - 8Y^4$

$Z3 \; = \; 2YZ$

$With\ A = 4XY^2\ and\ B = 3X^2 + aZ^4$

If we consider the case a = -3 .then computational cost is reduced to 5M+3S.

$$B = \frac{3(X - Z^2)}{(X + Z^2)}$$

Generally the computational cost of Point doubling in Jacobian co-ordinates is 5M+4S.

Level 1

2.6.2 Scalar Multiplication

This mathematical level deals with the efficient computation of dP *using* point operations introduced in the previous section.In the remainder of this thesis, we succinctly describe the most popular ones based on their efficiency in terms of speed and/or advantageous memory requirements.

The usual scalar multiplication is calculated by double and adds method. For example if 49P is calculated as

49P = 2(2(2(2(P+2P))))+P

Binary Method to Perform Scalar Multiplication

This is the traditional scalar multiplication based on the binary expansion of the scalar *d*using {0,1}. Given a binary representation of *d*, the scalar multiplication can be computedby scanning the bits of *d* from left to right,

Algorithm 2.1: Leftto Right Binary Method for scalar multiplication

Input : d = $(d_{l-1},\ldots,d_2,d_1,d_0)_2$ d ∈ $E(F_P)$
Output: $Q = dP$

```
Step 1   $Q = 0$
Step 2   for i=l-1 to downward 0
            2.1  $Q = 2Q$
            2.2  If $di = 1$ then
            2.3  $Q = Q + P$
Step 3   return ($Q$)
```

In Algorithm 2.1, the average number of doublings and additions are $l \approx n - 1$ and $l/2 \approx n/2$ respectively, as *d* tends to infinity. Thus the cost of the binary method is approximately as follows (n-1)D +(n/2)A

Example: Assume$d = 12632$ and P *is*a point on the elliptic curve E. Given the binary expansion of d:

- 12632 = $2^{13}+2^{12}+2^{8}+2^{6}+2^{4}+2^{3}$ =(11000101011000)2

  The scalar multiplication denoted by $[12632]P$ using above algorithm would be as follows:

- $[12632]P = 2^3(2(2^2(2^2(2^4(2P+P)+P)+P)+P)+P)$

2.7 ECC curves parameter

2.7.1 Domain Parameter of ECC over F_p

Elliptic curve over Fp has list of domain parameters which includes 'p', 'a',' b',' G', 'n' and 'h' parameters.

'a' and 'b': define the curve $Y^2 \, mod \, p = X^3 + aX + b \, mod \, p$.
'p': prime number defined for finite field Fp.
'G': Generator point (XG,YG) on the EC that selected for cryptography operations.
'n': The Elliptic curve order.
'h': if #E(Fp) is the number of points on an elliptic curve then 'h' is cofactor where $h = \frac{\#E(Fp)}{n}$

2.7.2 Domain Parameter of ECC over Binary Field

Elliptic curve over F_2^m has a list of domain parameters which includes 'm', f(x), 'a', 'b', 'G', 'n' and 'h' parameters.

'm': an integer to finite field F_2^m
F(x): the irreducible polynomial of degree m that it used for elliptic curve operations
'a' and 'b': define the curves $Y^2 + XY = X^3 + aX + b$
'G': the generator point (xG, yG) on the EC that selected for cryptography operations.
'n': the order of the elliptic curve.

'h': if #E(F_2^m) is the number of points on an elliptic curve then 'h' is cofactor where $h = \frac{\#E(F2^\wedge m)}{n}$

2.8 ECDH – Elliptic curve Diffie Hellman

ECDH, a variant of DH, is a key agreement algorithm. For generating a shared secret between A and B using ECDH, both have to agree up on Elliptic Curve domain parameters. An overview of ECC cryptographic algorithms for key agreement is explained here.

Key Agreement Algorithm

For establishing shared secret between two device A and B

Step 1: LetdA and dB be the private keys of device A and B respectively, Private keys are random number less than n, where n is an EC domain parameter.

Step 2: Let$QA = dA * G$ and $QB = dB * G$ be the public key of device A and B respectively, G is a domain parameter.

Step 3: A and B exchanged their public keys

Step 4: A computes $K = (XK, YK) = dA * QB$

Step 5 : B computes $L = (XL, YL) = dB * QA$

Step 6: Since K=L, shared secret is chosen as XK

ECDH - Mathematical Explanation

To prove the agreed shared secret K and L at both devices A and B are the same From Step2, Step4 and Step5.

$$K = dA * QB = dA * (dB * G) = (dB * dA) * G = dB * (dA * G) = dB * QA = L$$

Hence K = L, therefore XK = XL. Since it is practically impossible to find the private key dA or dB from the public key QAor QB, so it is impossible to obtain the shared secret for a third party.

2.9 NIST Recommended ECC curves for Cryptography

The principal parameters for elliptic curve cryptography are the elliptic curve E and a designated point G on E called the base point. The base point has order n, which is a large prime. The number of points on the curve is h*nfor some integer h (the cofactor), which is not divisible by n. For efficiency reasons, it is desirable to have the cofactor be as small as possible.All of the NIST recommended curves given in fig 2.7having cofactors 1, 2, or 4. As a result, the private and public keys for a curve are approximately the same length.

Choice of Underlying Fields

For each key length two types of fields are provided.

- A prime field is the field GF(p), which contains a prime number p of elements. The elements of this field are the integers modulo p, and the field arithmetic is implemented in terms of the arithmetic of integers modulo p.
- A binary field is the field GF(2^m), which contains 2^m elements for some m (called the degree of the field). The elements of this field are the bit strings of length m, and the field arithmetic is implemented in terms of operations on the bits.

The security strengths for five ranges of the bit length of *n* is provided in table 2.1. For the field *GF*(*p*), the security strength is dependent on the length of the binary expansion of *p*. For the field *GF*(2^m), the security strength is dependent on the value of *m*. The table given below provides the bit lengths of the various underlying fields of the curves. Column 1 lists the ranges for the bit length of *n*.Column 2 identifies the value of *p* used for the curves over prime fields, where len(*p*) is the length of the binary expansion of the integer *p*. Column 3 provides the value of *m* for the curves over binary fields.

Bit length of n	**Prime Field**	**Binary Field**
161-223	Len(p)= 192	m = 163
224-255	Len(p) = 224	m = 233
256-383	Len(p) = 256	m = 283
384-511	Len(p) = 384	m = 409
≥ 512	Len(p) = 521	m = 571

Table 2.4 Bit Lengths of the Underlying Fields of the Recommended Curves

Choices of Curves

Two kinds of curves are given:

- Pseudo-random curves are those whose coefficients are generated from the output of a seeded cryptographic hash function. If the domain parameter seed value is given along with the coefficients, it can be easily verified that the coefficients were generated by that method.

- Special curves are those whose coefficients and underlying field have been selected to optimize the efficiency of the elliptic curve operations.

Choice of Base Points

Any point of order *n* can serve as the base point. Each curve is supplied with a sample base point *G* =(*Gx*, *Gy*). Users may want to generate their own base points to ensure cryptographic separation of networks.

The parameter of NIST recommended curves is given below in table 2.5

$$p_{192} = 2^{192} - 2^{64} - 1$$

$$= 6277101735386680763835789423207666416083908700390324961279$$

$$n = 6277101735386680763835789423176059013767194773182842284081$$

$$a = -3$$

$$b = \text{64210519e59c80e70fa7e9ab72243049feb8deecc146b9b1}$$

$$h = -1$$

$$p_{224} = 2^{224} - 2^{96} + 1$$

$$= 26959946667150639794667015087019630673557916260026308143510066298881$$

$$n = 26959946667150639794667015087019625940457807714424391721682722368061$$

$$a = -3$$

$$b = b4050a85\ 0c04b3abf5413256\ 5044b0b7d7bfd8ba270b39432355ffb4$$

$$h = -1$$

$$p_{256} = 2^{256} - 2^{264} + 2^{192} + 2^{96} - 1$$

$$= 115792089210356248762697446949407573530086143415290314195533631308867097853951$$

$n = 115792089210356248762697446949407573529996955224135760342422259061068512044369$ $a = -3$ $b = 5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604b$ h = -1
$p_{384} = 2^{384} - 2^{128} - 2^{96} + 2^{32} - 1$ $= 39402006196394479212279040100143613805079739270465446667948293404245$ $721771496870329047266088258938001861606973112319$ $n = 39402006196394479212279040100143613805079739270465446667946905279627659399113263569398956308152294913554433653942643$ $a = -3$ $b = b3312fa7e23ee7e4988e056be3f82d19181d9c6efe8141120314088f\ 5013875ac656398d8a2ed19d2a85c8edd3ec2aef$ $h = -1$
$p_{521} = 2^{521} - 1$ $= 6864797660130609714981900799081393217269435300143305409394463459185543183397656052122559640661454554977296311391480858037121987999716643812574028291115057151$ $n = 6864797660130609714981900799081393217269435300143305409394463459185543183397655394245057746333217197532963996371363321113864768612440380340372808892707005449$ $a = -3$ $b = 051953eb9618e1c9a1f\ 929a21a0b68540eea2da725b99b315f3\ b8b489918ef109e156193951ec7e937b1652c0bd3bb1bf073573df883d2c34f1ef451fd46b503f00$ $h = -1$

Table2.5: NIST Recommended Elliptic Curves parameters

2.10 Elliptic Curves Discrete Logarithmic problem

Let E be the elliptic curve over the finite field Fp. We can represent the main operation inECC, namely scalar multiplication, as follows:

$$Q = d * P$$

Where P and Q are points in E(Fp) of order q, and d is the secret scalar. We define the Elliptic Curve Discrete Logarithm Problem (ECDLP) as the problem of determining scalar d, given P and Q.

It is believed that the usual discrete logarithm problem over the multiplicative group of a finite field (DLP) and ECDLP are not equivalent problems, and that ECDLP is significantly more difficult than DLP. The main reason is that there is no known sub exponential-time algorithm to solve ECDLP in general.

Security of systems based on ECC relies on the hardness of this problem. Ingeneral, ECDLP has proven to be harder than other recognized problems such as the integer Factorization problem and the discrete logarithm problem, which are the foundation of RSA (Rivest-Shamir-Adleman) and DH (Diffie-Hellman) cryptosystems respectively.

It is also possible to attack ECC using special-purpose hardware. Van Oorschot and Wiener [11] proposed an attack against a 120 bit EC system using special-purpose hardware. In their 1996 study, they estimated that if $n \approx 10^{36} \approx 2^{120}$, then a machine with r = 330,000 processors that could be built for about US $10 million would compute a single discrete logarithm in about 32 days. However, such hardware attacks are still infeasible for $n > 160$

2.11 ECC Attacks

There are two types of attacks, special-purpose and general-purpose, for solving ECDLP. Specialpurpose attack algorithms are tailored to perform better for the elliptic curves with a special form. In contrast, the running times of general-purpose attacks depend only on the size of elliptic curve parameters. In the next two sections we briefly overview some of the known general-purpose and special-purpose attack**.**

General-Purpose Attacks:

a. **Exhaustive Search:**
In exhaustive search, one attempts to solve the problem by trying all possible keys in the key space. This can be done by computing all successive multiplies of P, 2P, 3P, 4P,... . This method takes up to n steps, where n is the order of the point P.

b. **Baby-Step Giant-Step Algorithm:**This is a time-memory trade-off version of the exhaustive search method. It requiresstorage for about $\sqrt{n}$ points, and its running time is roughly $\sqrt{n}$ steps in the worst case.

c. **Pollard's Rho Algorithm**

This algorithm is a randomized version of the baby-step giant-step algorithm. It has roughly the same running time ($\sqrt{\frac{\pi n}{2}}$ steps) as the baby-step giant-step algorithm, but issuperior in that it requires a negligible amount of storage. Van Oorschot and Wiener [11] showed how Pollard's Rho algorithm can be parallelized so that when the algorithm is run in parallel on r processors, the expected running time of the algorithm is roughly $\sqrt{\frac{\pi n}{2r}}$ steps. At present, the parallelized version of Pollard's Rho algorithm is the fastest general-purpose method for solving the ECDLP.

d. **Multiple Logarithms**

Silverman and Stapleton [12] observed that if a single instance of the ECDLP (for a given elliptic curve and base point P) is solved, then the next instance for the same curve and the same base point can be solved more easily. More precisely, if solving the first instance takes expected time t, solving the second and third instances takes $(\sqrt{2}-1)$t and $(\sqrt{3}-\sqrt{2})$t respectively.

Special Purpose Attack:

- **Mov Attack :**

  Menezes, Okamoto and Vanstone (MOV) [13], showed how, under mild assumptions, the ECDLP in an elliptic curve defined over a finite field Fq can be reduced to the DLP in some extension field $F$for some b>1. This reduction is only useful when b is a small number (less than $\log^2$ (q)). Balasubramanian and Koblitz [13] showed that for most elliptic curves, b is not a small number. However, for a very special class of elliptic curves (known as supersingular curves), it is known that b≤6 . For these curves the MOV reduction gives a subexponential-time algorithm for solving ECDLP. For this reason supersingular curves are excluded from use in elliptic curve cryptosystems.

- **Prime Field Anomalous Attack :**

  Semaev [23], Smart [24], and Satoh and Araki [25] independently showed that it is easy to solve EDLP for a special class of elliptic curves called anomalous elliptic curves. An anomalous elliptic curve over F_q is an elliptic curve which has exactly q points.

- **Pollard's Rho Attack for Koblitz Curves:**
  Gallant, Lambert and Vanstone [17], and Wiener and Zuccherato [18] independently showed a way to speed up Pollard's Rho algorithm by a factor of$\sqrt{d}$for solving ECDLP for elliptic curves over Fq^{bd}. For example for a Koblitz curve over $F2^{m}$Pollard's Rho algorithm can be speed up by a factor of $\sqrt{m}$. In fact this factor is not a concern in practice since it is relatively small.

2.12 Countermeasures

Table 2.5 provides the known attack together with theircountermeasures. An elliptic curve that satisfies all the countermeasure requirements in this table is considered intractable against all known attacks.

Attack	**Countermeasures**
Pohlig–Hellman[12]	Select n to be Prime
Pollard Rho[12]	Select n to be large no (at least > 2^{160})
Multiple Logarithmic[12]	Select n to be large no (at least > 2^{160})
MOV[13]	Check n does not divide $q^{k} – 1$ for $1 \le k \le 20$
Prime field anomalous[23, 24]	Check that $n \neq q$
Weil Descent[25]	Do not use elliptic curves over composite binary fields or over F_{P}^{m} where P is odd and M is either 5 or 7.

Table 2.6 Summary of ECDLP attacks and their countermeasures

Summary: In this chapter we introduce Elliptic Curve Cryptography (ECC) and different types of co-ordinate system. Also we have defined basic mathematical operation of ECC (like addition, doubling, and scalar multiplication), co-ordinate conversion complexity, Elliptic curve discrete logarithmic problem (ECDLP), ECC attacks and counter measures of each attack.

Chapter 3

Analysis of attacks in aECC Based Protocol

3.1 Introduction:

The first step in the analysis of ECC based protocols which we consider in this thesis is to design a threat model. Threat modeling is a method for evaluating the security of a software application. It looks at a system from a possible attacker's mind-set. As shown in Fig. 3.1, to construct a threat model, it is necessary to specify the assumptions under which the protocols will be analyzed and various threats that pose security risks to the protocol.

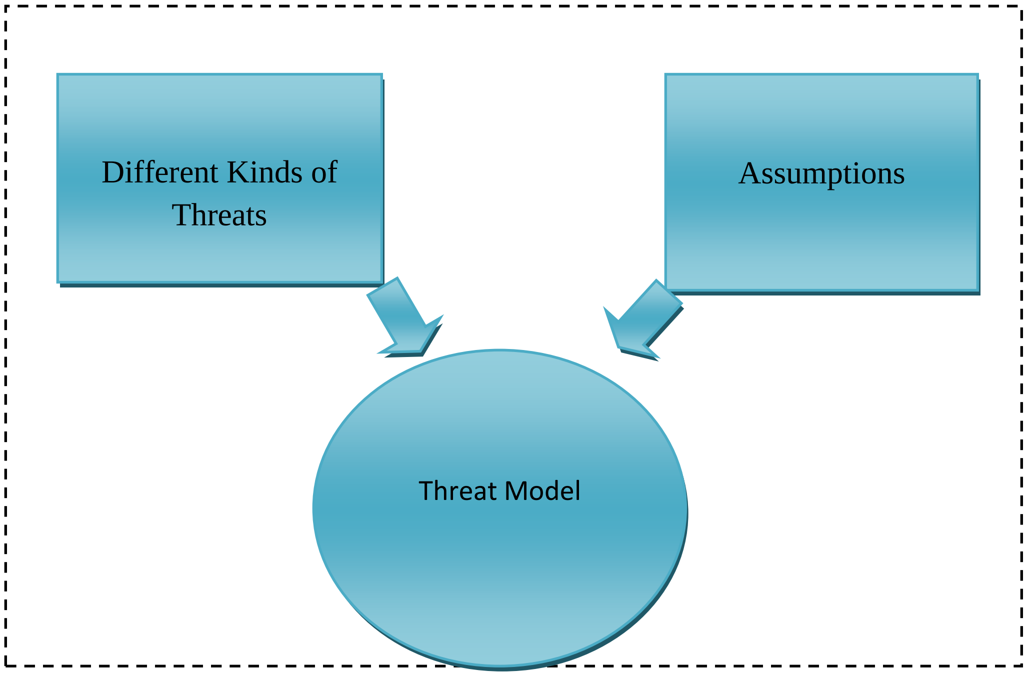


Fig 3.1 Threat Model

The next step is to choose authentication protocols to perform static cryptanalysis as shown in Fig. 3.2. This work focused solely on protocols that use ECC. Some of the other criteria used for selecting protocols include:

1. Protocol Recency
2. Variation in usage of authentication factors (e.g. smart cards, RFIDs, memory drives, etc.)
3. Variation in techniques to implement security (e.g. timestamps, nonce, encryption, hashes, etc.)

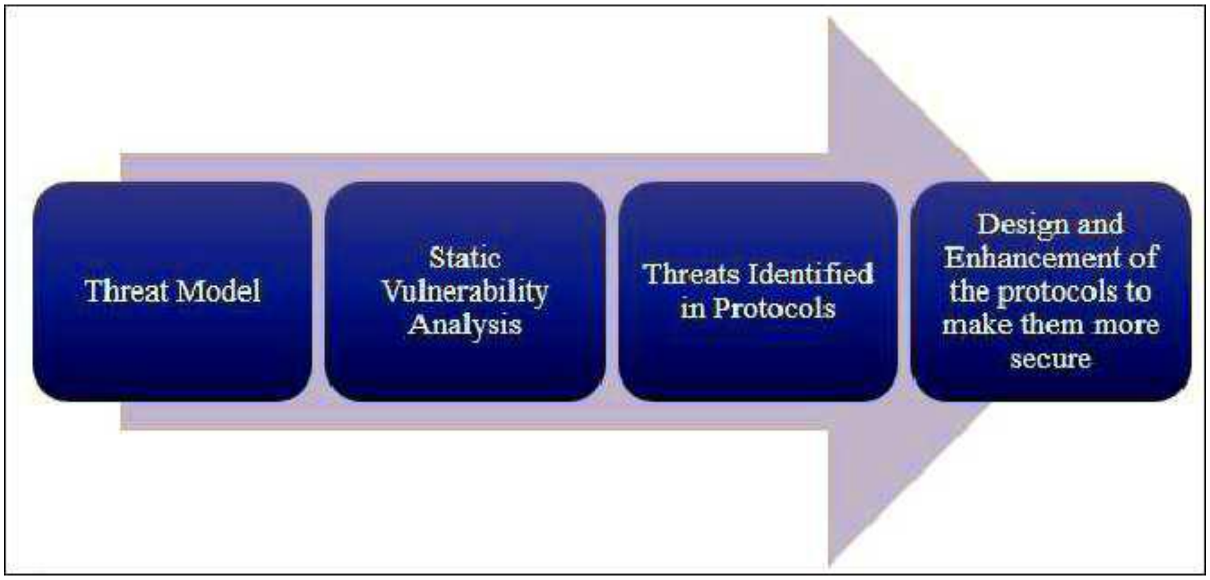

Threat Model
Static Vulnerability Analysis
Threats Identified in Protocols
Design and Enhancement of the protocols to make them more secure

Fig 3.2 Threat Model Analysis Steps

Threat vulnerability analysis was performed on the protocols identified to find whether they are indeed vulnerable to the threats. The last step is to suggest solutions to prevent the identified attack in order to make the protocols more secure.

3.2 Application of the Threat Model to Analyze Protocols:

A threat model helps to assess the probability, potential harm, and priority of attacks on a given ECC based protocol, and thus helps to minimize threats in the protocols. It is often useful to define many different threat models for a system (of protocols), with each model representing a different set of analysis, where each set contains different types of vulnerabilities. Threat identification is intended to identify potential threats in system components that might lead to a breach in the overall security of a system. The absence of security against a threat could denote a vulnerability whose risk could be reduced with the application of a countermeasure. Threats in protocols are identified by performing vulnerability analysis and finding flaws in protocol design. Analysis results are used to suggest improvements to the protocol to prevent possible attacks and make it more secure.

3.3 General Algorithms and Conditions for Various Attacks:

In this section, general algorithms and conditions for the attacks that were successfully performed on the protocol considered for this work are described.

3.3.1 Clogging Attack

The technique for almost all password authentication protocols is that the client (often a smart card reader, memory stick, or RFID) provides its authorization to the server, which in turn

computes particular arithmetic operations in order to validate the credentials. These protocols often function in multiple phases.
As shown in Fig. 3.3, the prime concept of the clogging attack is blockage of the message that contains login credentials between the client and the server [17]. This message is not encrypted in a few protocols and encrypted in others. It may or may not contain time stamp.

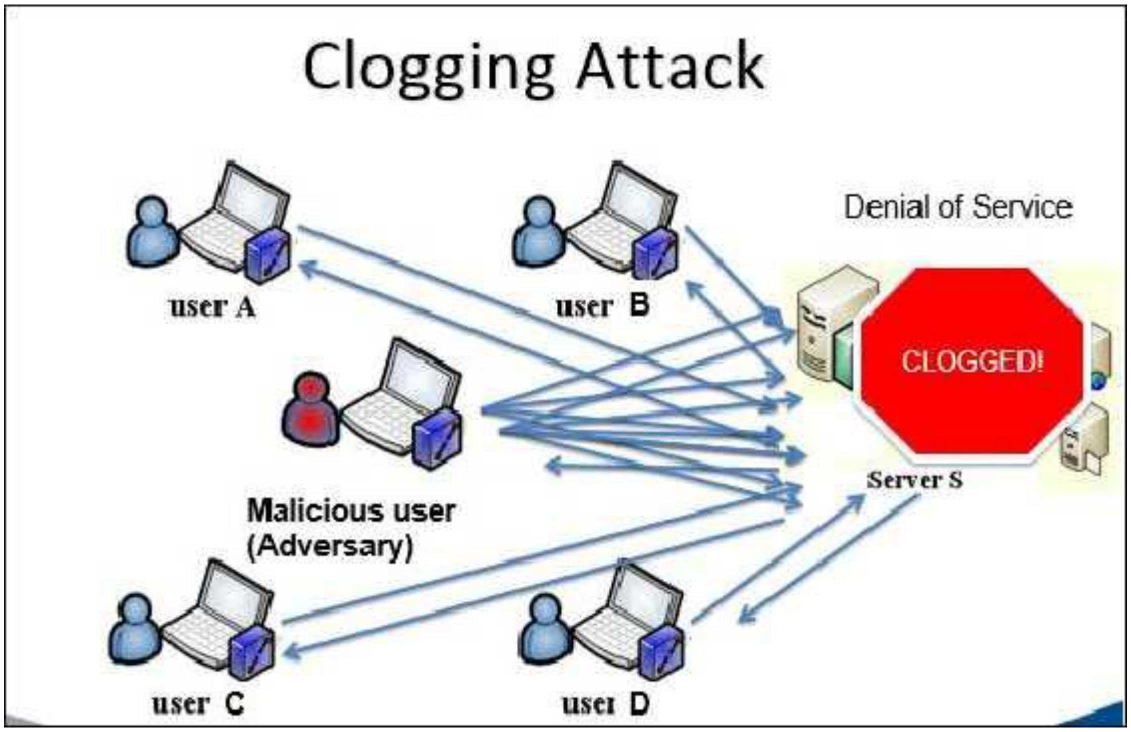


Fig 3.3: Clogging Attack

The attacker replays the intercepted message multiple timesto enforce the server to carry out computationally intensive ECC operations,[18] hence enforcing the server to lose its time and resources. Authorized users are blocked services in this way. Algorithm for this type of attack is as follows.

The **Algorithm3.1** to perform Clogging Attack:

```
Intercept login message from client to server
IfTimestamp is present then
Change timestamp to suit requirements
Else
Keep message as is
End if
While The server is not completely clogged! do
Replay the message to the server
end while
```

The clogging elements are evaluated in this work which relies on the computational and resource intensiveness of the operations in elliptic curve cryptography (ECC). The widely used ECC operations by most of the authentication schemes are:

1. Bilinear Pairing
2. Scalar Multiplication in Group G
3. Map to Point Conversion

Let Tp, Ts, and Tmapbe the time taken to perform a single bilinear pairing, scalar multiplication, and map-to-point conversation respectively. It has been shown in[19]

1. $T_P > T_S > T_{map}$
2. $T_P \approx 3T_s$
3. $T_P \approx 4T_{map}$

Further, let Tmodexbe the time taken by one modulo exponentiation operation. It has been proven [20] that Tp≈ 2×Tmodex for the same level of security. The operation modular exponentiation has been shown to be very computationally intensive [17].In fact, Tmodexhas been shown to be approximately a hundred times that of normal addition, multiplication, and bitwise XOR operations. We can, hence, conclude that all the ECC based operations bilinear pairing, scalar multiplication, and map-to-point conversation are computationally intensive.

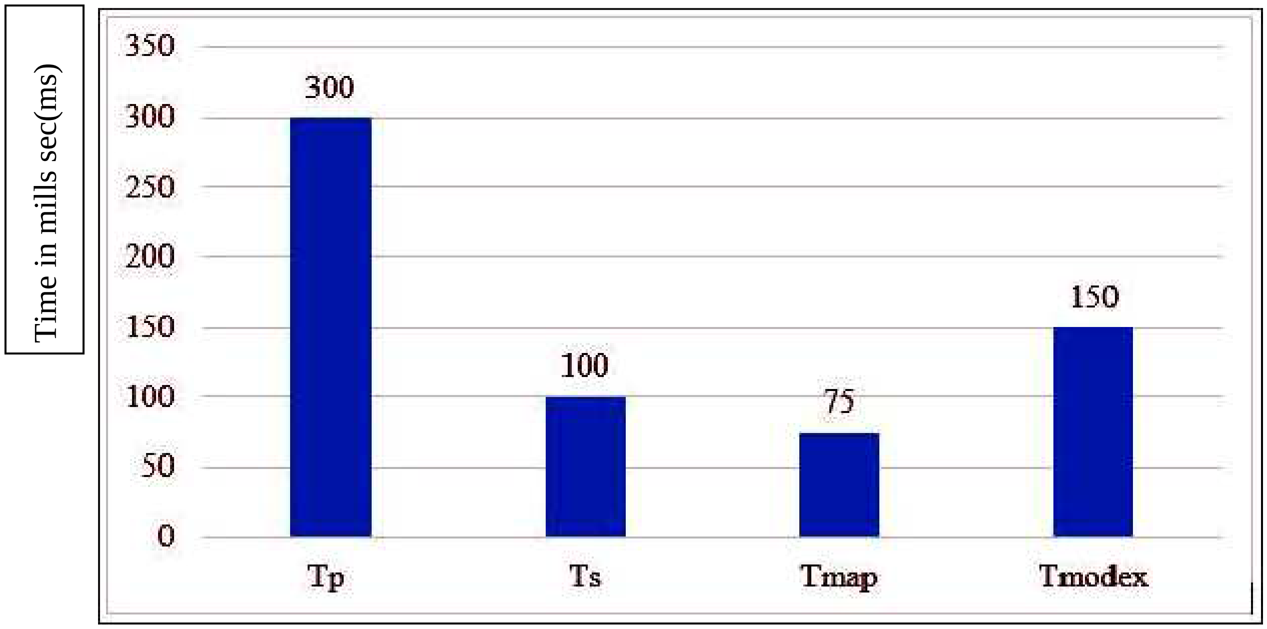


Fig 3.4 Comparison of ECC and modex operation

Hence, a protocol that uses ECC operation has a vulnerability to the clogging attack, a type of DoS in which the attacker abuse the computational intensiveness nature of ECC operations.

3.3.2 Data Base Attack:

According to many experts, databases are still not secured properly in most organizations [21]. Database attacks go unnoticed as it takes less than a few seconds to hack in and out of a database. Therefore, it is not a wonder that a lot of database attacks go undiscovered by large organizations until late after the information has been compromised. Attackers use clean methods to cause a breach in databases, such as exploiting weak authentication, using default passwords and exploiting familiar vulnerabilities [21].

This analysis focuses on database connections which are weak and hence open to vulnerabilities. The front end client-server authentication stores passwords in the server's back-end databases. If any password is compromised, then the database schema becomes vulnerable to attack which makes the protocol insecure (as explained in Algorithm 3.2). Passwords and their hashed forms are usually stored in relational databases. The most familiar approach to get unauthorized access to a database is to make a copy of the database by a technique called SQL injection. SQL injection attacks appears where the fields accessible for user input allow SQL statements through to query the Data base instantly. Web applications generally are the weakest link outside of the client's architecture [21].

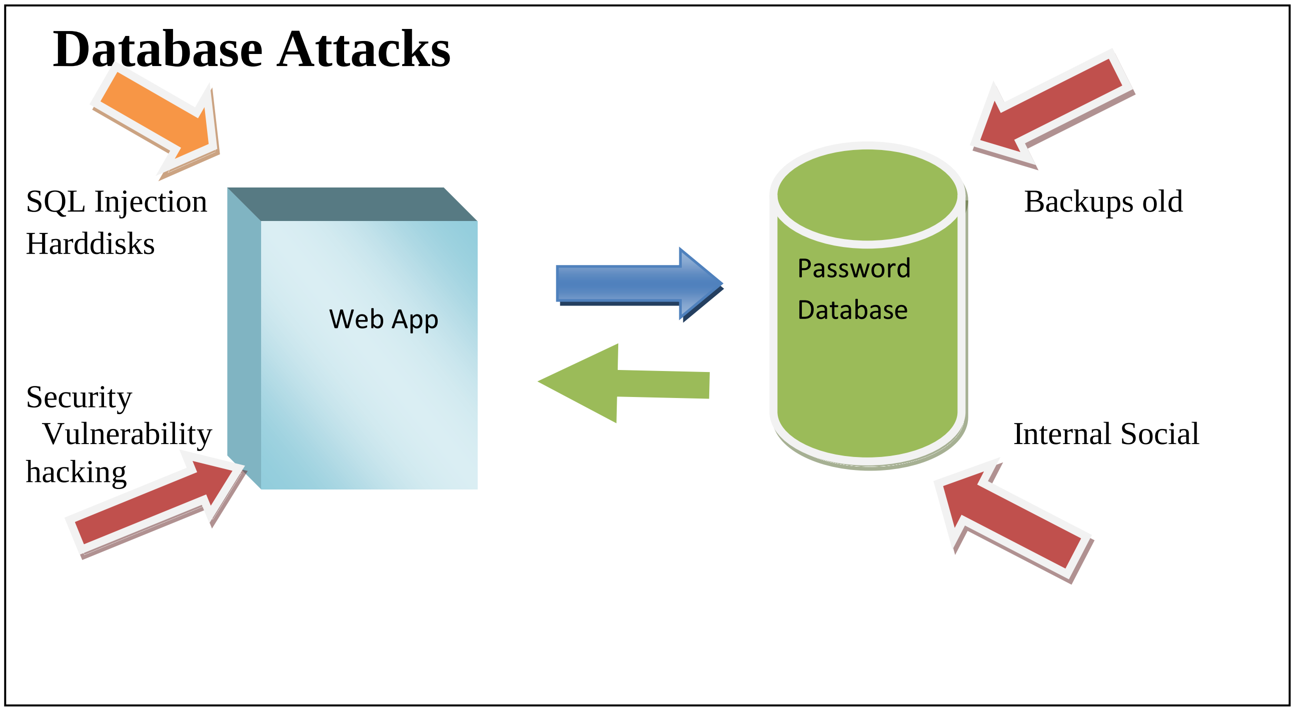


Fig: 3.5 Data Base Attack

Internal attacks should also never be underestimated. There have been many cases of insider attacks which came as a result of a malicious user acquiring more system access than the user should have had [21].Databases are usually attainable from inside organizations and passwords can easily be found in the source code or configuration files. This gives an opportunity to employees to access data and save it to a local disk or even transfer it to an external output device.

Algorithm 3.2: General Algorithm for Data Base Attack

```
Intercept data access layer from application to back-end
IfEncryption is present then
   Break the encryption to gain access to the database
else
  Access the database
end if
whileThe data are not corrupted and stolendo
  Inject malicious statements
end while
```

3.3.3 Man in Middle Attack

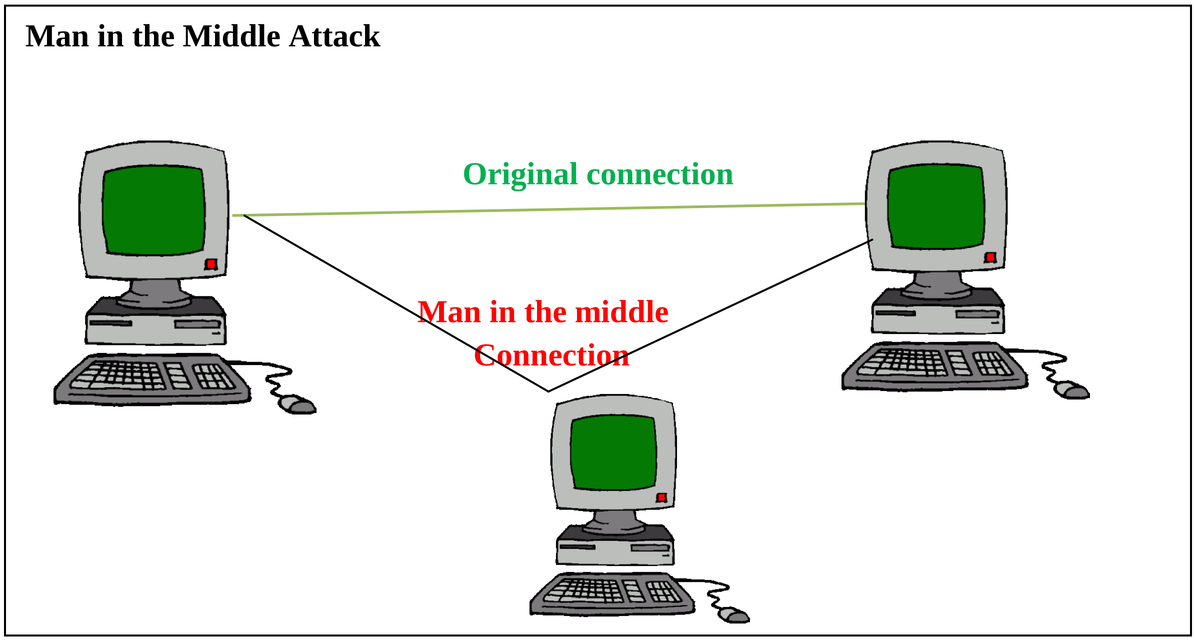


Fig 3.6: Man in middle Attack

As shown in Fig.3.6, a man-in-the-middle attack can be used towards some of the cryptographic protocols. A man-in-the-middle attack needs an attacker to gain the capability to control and inject messages onto a communication medium. One example is eavesdropping, wherein the attacker attempts to make separate communications with the victims and relays messages between them to make them believe they are talking with each other over a private network, where as in reality the complete conversation is governed by the attacker. Attacker has the ability to intercept all the messages passing between the two victims and inject new ones. A few ECC based protocols claim to be secure against man-in-the-middle attacks. However, ECC protocols are still vulnerable to man-in-the-middle attacks.

Algorithm 3.3: A general algorithm for man in middle attack

```
 Intercept communication between two parties
IfTTP is present then
 Acquires access and potentially alters the communication between two victimswho
are bound to believe they are directly communicating with each other
else
 Acts as an invader who relays and modifies the message between two victims
end if
whileThe communication is not ended do
 Relay
end while
```

3.4 Basics of RFID

Radio-frequency identification (RFID) is an automatic identification technology that transmits data through the use of wireless communication using radio waves. The first use of RFID system was in the World War II for the friend or foe? Identification system.

An RFID system basically consists of three components:

1. An antenna
2. A transceiver (with decoder)
3. A transponder (or RFID-tag)

The **antenna** is usually integrated in the transceiver and decoder making it a reader. Radio signals are emitted by the antenna activating the tag and reading or writing data on it. The antenna produces an electromagnetic field which can either be permanent when large numbers of tags are expected continuously, like on freeway toll booths, or it can be activated and deactivated whenever needed.

A **transponder,** or RFID-tag, is the “data-carrying device” of an RFID-system and normally consists of a coupling element and an electronic microchip.It usually does not have an own battery so it is passive as long as it is not in the interrogation zone of a RFID reader. The power needed to send and receive data as well as to change data is sent by the reader through the coupling unit [27].The tags have to be classified into read/write and read-only tags. The data on read/write tags can be altered or rewritten. Read-only tags, in contrast, are programmed with a unique set of data, which is usually 32 to 128 bits that cannot be modified.

Furthermore, RFID-tags are either active or passive. Tags powered by an internal battery are called active and usually read/write whereas passive tags are not equipped with an internal energy source but are powered by the reader. The main differences between those two kinds of tags are the following: Passive tags are much lighter, cheaper and have an unlimited operational lifetime but they have shorter read ranges and require a higher-powered reader.

Another distinction is the frequency range of the radio waves. “Radio frequency, or RF, refers to that portion of the electromagnetic spectrum in which electromagnetic waves can be generated by alternating current fed to an antenna.RFID runs at different bands and frequencies within the bands. This is caused by the different applications required. The table 3.1 [28] below shows the various frequencies, which are used for the different applications of RFID. Furthermore, the advantages and disadvantages of the frequencies are listed as well as applications.

Band + Frequency	**Read range**	**Advantages**	**Application**
Low Frequency (LF) 30-300 KHz	Up to 20 inches (~0.5m)	+ Good penetration in moist environments + No Anticollision - Slow data rate	• Animal Tagging (134.2 KHz), • Access control • Vehicle key locks,..
High Frequency (HF) 3- 30 MHz	Up to 3 feet (1m)	+ Good penetration in moist environments - Poor performance in metal environment	• Item level tagging, libraries • Smart cards,.. • Airline baggage
Ultra High Frequency (UHF) 300-3000 MHz	Passive: Up to 16 feet. Active: More than 30 feet (6m)	+ Fast data rates + Good performance in metal environment	• Supply chain use at WM and Metro • Baggage handling • Toll collection, ..
Super High Frequency (SHF) 3-30 GHz	2+ meters	+ Fast data rates + Good performance in metal environment - Poor performance in moist environment - High cost	• Item tracking • Toll collection

Table 3.1 RFID-Frequencies

Chapter 4

4.1 Our Model

Model Description:

The main aim of our model is to design light weight efficient scalar multiplier which can be embedded into an ECC based light weight protocol. We have used NAF algorithm to convert an integer K in to NAF sign form. After that this signed K is used in a scalar multiplication algorithm which is computationally as well as memory efficient than traditional method. We know that the inversion cost is high. To further reduce computational cost we must eliminate inversion operation and we have selectedinversion free co-ordinate system. There are different types of co-ordinate systems; in some co-ordinates system addition cost is too high while doubling cost is low and the reverse is true for some coordinates also. It is impossible to achieve minimum computational cost for doubling as well as addition operations using a single co-ordinate system. For efficient addition operation we considered mixed co-ordinates Jacobian and affine co-ordinates and result is in Jacobian co-ordinate. Similarly for doubling operation we used modified Jacobian co-ordinate. We know that ECC based scalar multiplication is performed by doubling and addition operation, so we used this efficient doubling and addition operation for scalar multiplication and verified using python programming language that proposed scalar multiplier is computational efficient. We can further improve the computational time of scalar multiplication but for that we need extra memory. In Light weight IoT authentication protocol based on ECC if we embedded the proposed scalar multiplication then the overall computational cost of the protocol will be minimized. Suggested scalar multiplication is embedded into an ECDSA and verified that the signature generation and verification time is less compared to the existing scheme.

4.2 Flow Chart of Our Model

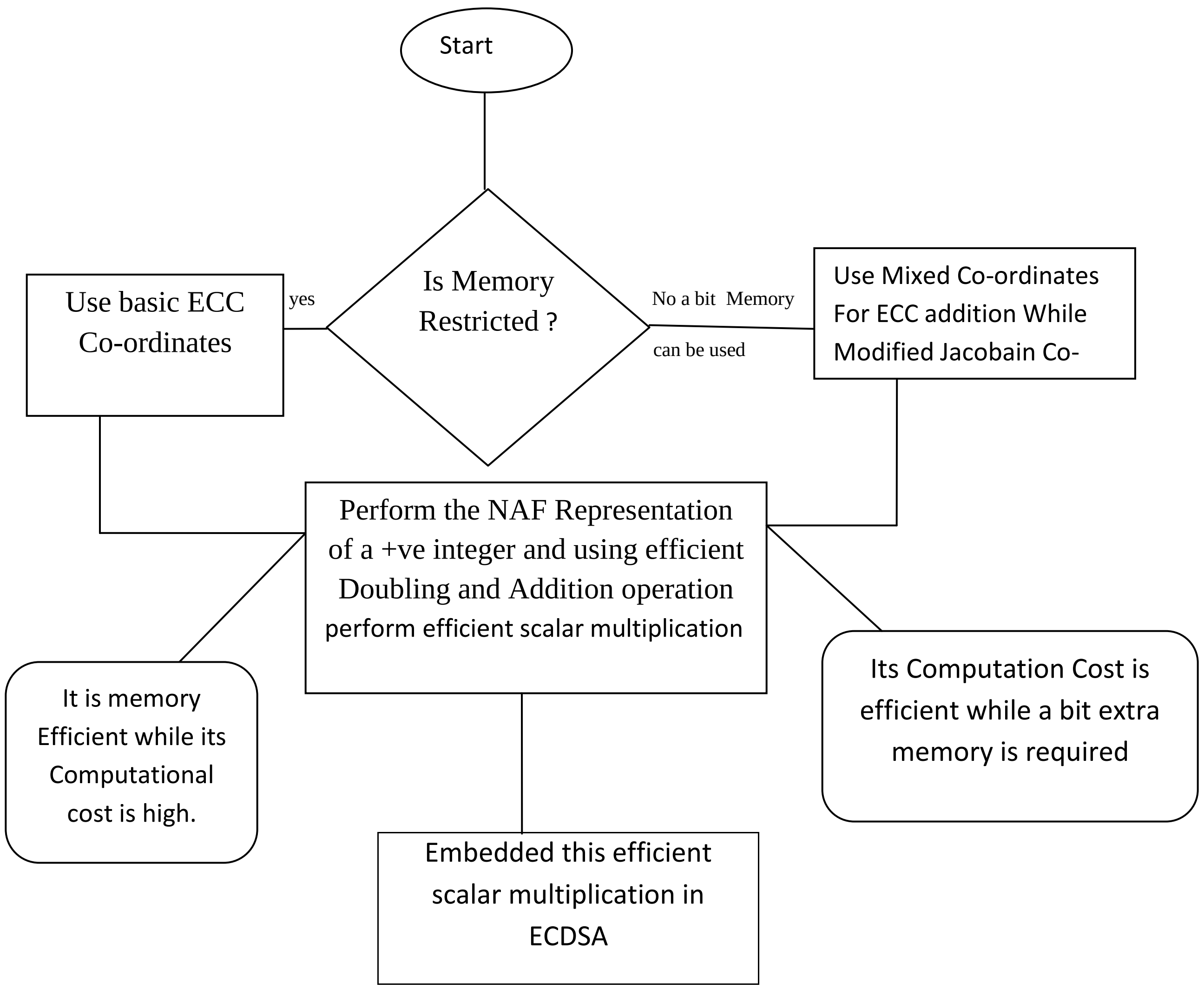

Start
Is Memory Restricted ?
yes
No a bit Memory can be used
Use basic ECC Co-ordinates
Use Mixed Co-ordinates For ECC addition While Modified Jacobain Co-
Perform the NAF Representation of a +ve integer and using efficient Doubling and Addition operation perform efficient scalar multiplication
It is memory Efficient while its Computational cost is high.
Its Computation Cost is efficient while a bit extra memory is required
Embedded this efficient scalar multiplication in ECDSA

Fig 4.1 Flow chart of proposed work

4.3 Elliptic Curves Digital Signature Algorithm

The figure 4.2 shows that in the given algorithm hash function is used. As shown in the figure, the digital signature algorithm takes a message digest instead of the message as the input .This is because the message digest is small compared to the message itself. Furthermore, a message digest can be made public since it does not reveal the contents of the message from which it is derived.

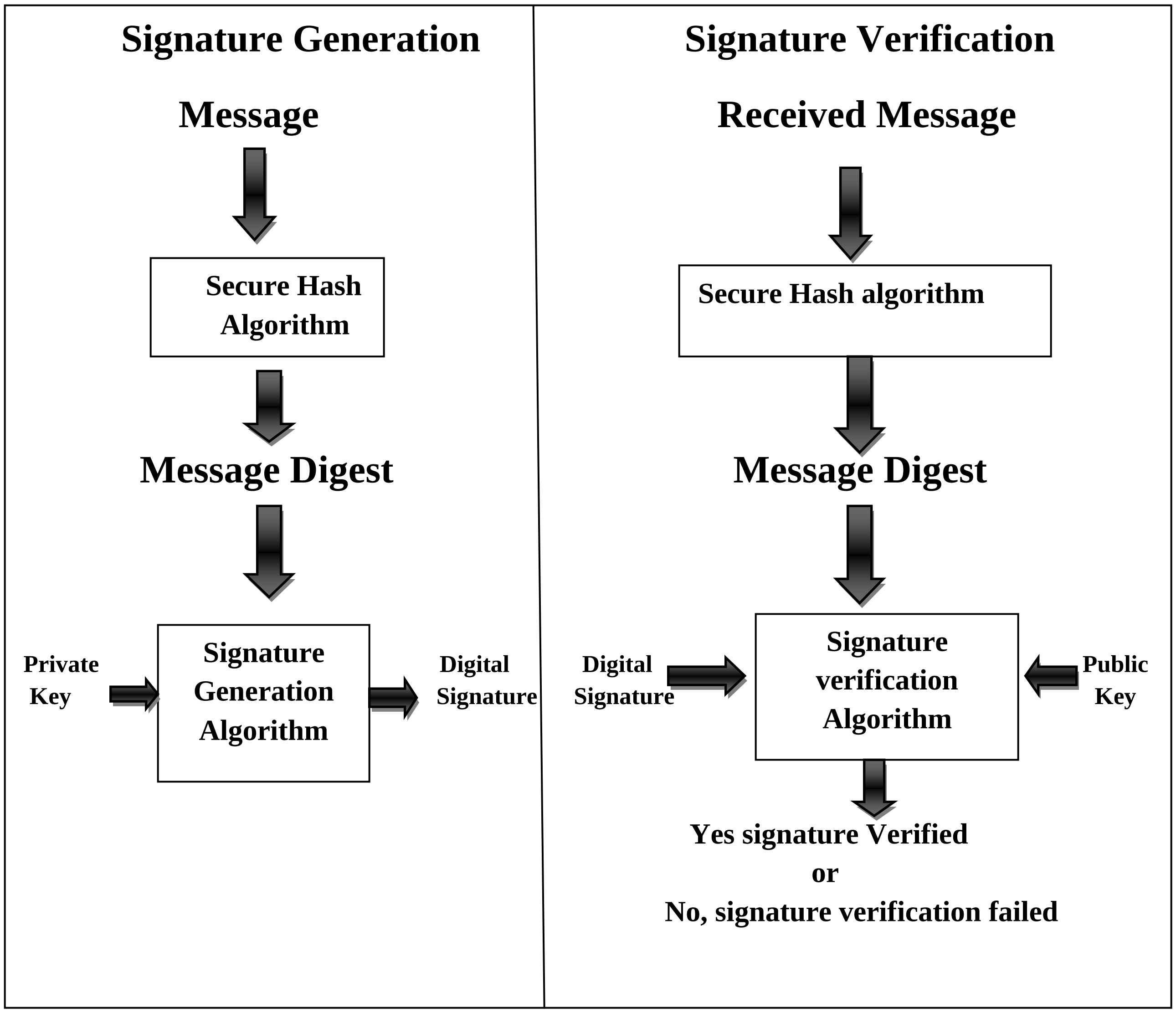


Fig4.2 Block Diagram of Signature Generation and Verification Process

The Elliptic Curve Digital Signature Algorithm (ECDSA) is a variant of the Digital Signature Algorithm [22] (DSA) which operates on elliptic curve groups. In order to sign a message, entity A needs to make public the system parameters such as:

- Prime Field p
- Elliptic curve E $_{a, b}$
- Base Point P
- Order n of the Base Point.

Let the integer d, d ∈ [1, n - l], be A's private key and the point $Q = d * P$ its public key. In order to sign message M, A does the following.

1. Generate a random no k ∈ [1 , n-1]
2. Compute $R = k * P$
3. Compute $r = x \, mod \, n$ where x is the x -coordinate of R. If r = 0 then go to step 1
4. Compute $e = H(M)$, where H is a cryptographic hash function such as SHA-1
5. Compute $s = k^{-1} (e + d * r) mod \, n$. If s = 0 then go to step 1
6. The signature on $M \, is \, (r, s)$

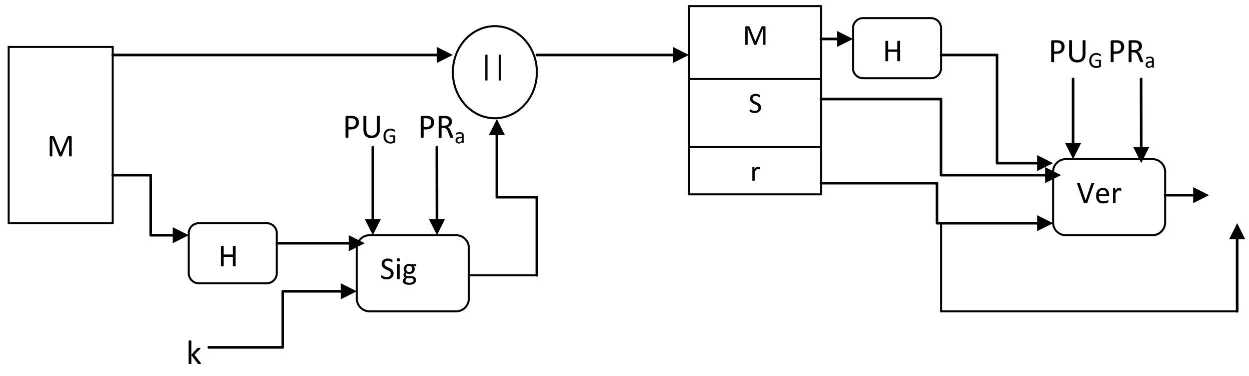


Fig 4.3 ECDSA Block Diagram

The entity B can verify A's signature $(r, s) \; on \; M$ as follows:

1. Verify that r and s are integers in the interval [1, n - 1]
2. Compute $e = H(M)$, where H is the same hash function used for generating the signature
3. Compute $w = s^{-1} \, mod \, n, u1 = e * w \, mod \, nand \, u2 = rw \, mod \, n$
4. Compute $X = u1 * P + u2 * Q$, and verify that X ≠0
5. Let $v = x \, mod \, n$, where x is the x-coordinate of X
6. Accept the signature if and only $if \; v = r$.

The requirement $v = r$ holds because if the signature (r, s) on a message M is indeed generated by A then.

$$k = s^{-1}(e + d * r) = s^{-1} * e + s^{-1} * r * d = w * e + w * r * d$$
$$= u1 + u2 * d \ (mod\ n)$$

And thus

$$X = u1 * P + u2 * Q = (u1 + u2 * d)P = k * P <=> v = r$$

The NAF algorithm

Many algorithms have been proposed to compute kP, where$P \in (Fp)$. These algorithms mainly depend on the recoding method of exponent k. The most popular method for performing EC point multiplication of the form $kP$is the Double-and-Add(Binary) method which uses the digit set D_W= {0,1} to represent exponent k. The efficiency of this method can be further improved using signed binary representations.The signed method use the digit set D_W ={0,1} .Also Further improvement can be achieved if some pre computational are allowed such as window recording and multiplication technique where the digit set D_w includes the more value

$D_W = \{0, \pm1, \pm2, \pm3, \dots\}$

The Non-Adjacent Form (NAF) is a signed binary representation of an integer with B = (1,-1). In this representation, at most one of any two consecutive digits is nonzero. Moreover, each positive integer has a unique NAF representation with expected weight $l/3$. Thus adding a negative digit to binary representation reduces the average density from $l/_2$to $l/3$.

Computing NAF of a any positive integer K

Input:A +ve integer K

Output : $NAF(K)$

```
1. i = 0
2. while k≥1 do
   2.1 if k is odd then $ki = 2 - k(mod)4$ , $k = k - ki$
   2.2 $else\ ki = 0$
   2.3 $k = k/2$
   2.4 $i = i + 1$
3. Return $K = (K_{i-1}, \dots K_2, K_1, K_0)$
```

4.4 Accelerating ECC Operation.

In this section, we investigate a new strategy for elliptic curve exponentiation. Up to now, since only one kind of coordinate system is used, it has been necessary that it should offer both an addition and a doubling with reasonable speed (not the fastest but not too slow). TheChudnovskyJacobian coordinate system is a good example: it reduces the computation time of an addition by slightly increasing the doubling time, but this is still worthwhile since Jacobian coordinates have a rather faster doubling but slower addition times than projective coordinates. On the contrary, here we further improve on the Jacobian coordinate system in order to offer even faster doublings, and there will be no loss in elliptic curve exponentiation since we are going to use a new strategy of mixed coordinate systems.

Here we modify the Jacobian coordinates in order to obtain the fastest possible doublings. For this, we represent internally the Jacobian coordinates as a quadruple (X, Y, Z, aZ4). We call this the modified Jacobian coordinate system, and denote it by J^m.

ECC Doubling using Modified Jacobian Co-ordinates:

Let $P = (X, Y, Z, aZ^4)$ then $R = 2P = (X3, Y3, Z3, aZ3^4)$

$$X3 = T$$

$$Y3 = M(S - T) - U$$

$$Z3 = 2Y1Z1$$

$$aZ3^4 = aZ3^4$$

Where$S = 4X1Y1^2$,$U = 8Y1^4$, $M = 3X1^2 + aZ1^4$, $T = -2S + M^2$

The computation time is $t(2J^m) = 4M + 4S$

ECC Addition Using Mixed Co-ordinates:

It is evidently possible to mix different coordinates, i.e. to add two points where one is given in some coordinate system, and the other point is in some other coordinate system.Since we have five different kinds of coordinate systems (represented by the symbols A, P, J,J^c, and J^m), this gives a large number of possibilities. The List of possibilities are given in a table 4.1.

Computational cost of ECC addition in different co-ordinates	
ADDITION OPERATION	**COMPUTATIONAL TIME**
$t(J^m + J^m)$	$13M + 6S$
$t(J + J)$	$12M + 4S$
$t(P + P)$	$12M + 2S$
$t(J^c + J^c)$	$11M + 2S$
$t(A + A)$	$2M + 2S + I$
$t(J^m + A = J^m)$	$9M + 5S$
$t(J + A = J)$	$8M + 3S$

Table 4.1 Computational Cost of ECC addition in different Co-ordinates

Note: The Symbols Representation is given following

J^m = Modified Jacobian co-ordinate

J = Jacobian co-ordinate

P = Projective co-ordinate

A = Affine co-ordinate

M = Multiplication operation

S = Squaring operation

I = Inverse operation

$t(c1 + c1)$= Computation time for addition operation in c1 co-ordinate and result in also in c1.

$t\,(c1 + c2 = c1)$ = Computation time for addition operation in c1 and c2 co-ordinate while results in c1 co-ordinate.

If we want to compare computation times,The ratio S/M is almost independent of the field of definition and of the implementation, and can be reasonably taken equal to 0.8

On the other hand, the ratio I/M deeply depend on the field of definition and on the implementation.It can be estimated to be between 9M and 30M in the case of p larger than 100 bits.

4.5 Final Algorithm of ECC Scalar Multiplication:

Input: $K \geq 0,\ P = (X, Y)$

Output: $Q = KP$

```
Step1 Convert K to NAF Form From above given algorithm $K = NAF(K)$
Step2   Find length of K
Step3   $Q = P$
Step 4   while ($length - 2 \geq 0$) do
        4.1  $Q = 2Q$
        4.2  $If\ Ki\ ==\ 1\ then\ Q = Q + P$
        4.3  $If\ ki\ == -1\ then\ Q = Q - P$
Step 5   Return $Q$
```

4.6 Moosavi et al.'s Protocol for RFID Implant Systems

The first protocol considered is that of Moosavi et al., 2014. This is a mutual authentication scheme for an RFID implant system. (Moosavi et al, 2014) assert that their protocol is immune to various attacks including denial of service (DoS). But, their protocol is inherently vulnerable to clogging attacks (a form of DoS). Most of the precursor protocols to that of (Moosavi et al., 2014) are vulnerable to clogging attacks. In this section, the mathematical groundwork that makes the protocols vulnerable to clogging attack is identified, and a desirable countermeasure is suggested.

Review of the Protocol:

Moosavi et al.'s protocol works in three phases, Reader Authentication and Verification, Tag Identification and Tag verification. The notation of the protocol is given below.

G: Elliptic curve generator point of field q and order n

P: A primitive element or the base point of G

$S1, S2$: Tag secret points S1, S2 ∈E(Fq), which will change over time.These secret point is varied with time and the tag is successfully identified.
IDt: The tag identity number or ID
S3:Each reader keeps a secret point S3 ∈ Zn. Which will change over time .this secret point will be varied each time after the reader is successfully authenticated.
IDr: $S3 * P$ thereader's public key.

r_s , i_1, i_2: Random number that ∈ Zn.

h: It is a light weight hash function.

(d, c): It is a generated signature by tags during its identification phase.

This protocol allows the two interacting parties, an RFID implant tag and a reader, toRespectively validate and assure both identities. The assumption is that the communication between the reader and tag is weak [18].

Analysis of Moosavi et al.'s Protocol

This protocol is for an RFID implant systemthat has applications inmicrochip implant.

The identities are the tag that is implanted, and the reader that verifies (and authenticates) the tag. Communication between the tag and the reader is through an insecure network. Additionally, the reader is connected to a database through a secured channel, so the reader and database is considered to be a single entity for analysis purpose [18]. The protocol uses ECC techniques twice, once during tag Identification (step I4 of Algorithm4.1) and once during tag verification (step V6 of Algorithm4.1)

Algorithm 4.1: Moosavi et al.'s Protocol for RFID Implant Systems

---

Reader Authentication and Verification

Reader R

1. **Step A1** Select a random number $r1 \in \boldsymbol{Zn}$and computes $R1 = r1 \cdot P$ as its public key
2. **Step A2** Initialize$i1$ to 1
3. **Step A3** $R \rightarrow T$: $\{R1, i1\}$

4. **Step A4** Increment$i1$ by $r1$

Tag

1. **Step A5**Verify if $i1 \geq i2$ ($i2$is initialized to 0)
2. **Step A6**If the above is true, then set$i2\ to\ i1$
3. **Step A7**Compute $r3 = X(r2 \cdot P) * Y\ (R1), where\ *$is a non-algebraic operation over the abscissa of $(r2 \cdot P)$ and the coordinate of $R1$

4. **Step A8**$T \rightarrow R: \{r3\}$

Reader R

1. **Step A9**Compute $R2 = r1 \cdot IDt + r3 \cdot s3$
2. **Step A10**$R \rightarrow T : \{R2\}$

Tag T

1. **Step A11**Verify $if(R2 - r1.IDt)r3^{-1}.P = IDr$
2. **Step A12** Reader*R*gets verified if the above is true

Tag Idendification

Reader R

1. **Step I1**Select $rs \in \boldsymbol{Z}n$, a random integer
2. **Step I2** $R \rightarrow T : \{rs\}$

Tag T

1. **Step I3** Validate if $rs \neq 0$. If success, then compute $s2 = f\ (X(s1)) \cdot P$
2. **Step I4** Select a random integer k and calculate curve point $(x, y) = k \cdot G$
3. **Step I5**Calculate $d = x\ mod\ n$
4. **Step I6**Validate if *d* = 0. If success, recalculate *d* using a different *k*
5. **Step I7**Calculate value of ID as $IDt = (Mb(X(s1)) * Mb(X(s2))) \cdot P$
6. **Step I8** Calculate $c = k \cdot (Hash(IDt) + X(s1) * d)\ mod\ n$
7. **Step I9** Validate if *c* = 0. If yes, recalculate *c* using a different *k*
8. **Step I10** $T \rightarrow R: \{IDt, (d, c)\}$

Tag Verification

Reader R

1. **Step V1**Select a random integer $rs \in \boldsymbol{Zn}$
2. **Step V2** Compute public key $pr = rs \cdot P$
3. **Step V3** Verify if $d, c \in Zn$
4. **Step V4** If true, compute $h = Hash(IDt)$
5. **Step V5** Compute $w = c^{-1} \, mod \, n, u1 = zw \, mod \, n, and \, u2 = dw \, mod \, n$
6. **Step V6** Compute curve point $(x, y) = u1 \cdot P + pr$
7. **Step V7** Verify if $r = x \, mod \, n.$ If true, authenticate tag T

4.6.1 Clogging Attack on Moosavi et al.'s Protocol

Consider the adversary A has the same power as assumed by Moosavi et al. Adversary Aneeds to be able to read and modify the contents of messages over a not so secure medium during the Tag Identification and Tag Verification phases of the protocol. The line of attack in this work is a denial of service where the aim of Adversary is to damage the whole RFID system.There are two steps (a map-to-point conversion in Step V6, and a scalarmultiplication in Step V2) where ECC schemes are applied that Adversary can try [18]. However, it might bemore profitable to render the Reader *R* useless; the line of attack selected here will try breaking the Reader *R* in a way that it will block services to authentic tags.

1. Adversary A intercepts an authentic message of $T \rightarrow R$: $\{IDt, (d, c)\}$from step I10.
2. As the message is not encrypted, Adversary can always modify the (d, c) such that $d, c \in Z_n$holds(though Adversary might not need to).
3. Adversary simply relays $\{IDt, (d, c)\} \; to \; R$

The reader perform step V1 to V7 that are mention in protocol.Adversary would make the reader *R* re-run steps V1 through V7 to calculate the ECC operations many times. Adversary has the potential to modify the incoming login requests from an authorized tag to *R*.As the ECC operations are computationally intensive [20], the victimized Reader *R* spends considerable computing resources performing unwanted ECC computations (amap-to-point conversion in Step V6, and a scalar multiplication in Step V2) along with the other steps V1, V3 through V5, and V7 rather than any real work. Thus adversary clogs *R* with unwanted work and hence denies an authorized tag (user) any service.Adversary needs only an ID of a single authentic tag to implement the clogging attack [18].

4.7 Proposed Counter Measure from the Attack

There are two ways to provide the solution of clogging attack that are present in the protocol. Initially in authentication phase reader may validate to see that the network address of the tag is authentic.It has to learn the network addresses of all the registered authentic tags.Adversary could still deceit the network address of a authentic tagand replay the tag verification message.To prevent this deceit, a cookie exchange stepmay be inserted at the beginning of the tag verification phase of Moosavi et al.'s protocol [18], or we can also eliminate this attack by encoding the tag credential information in step I10 by tag private key while other side it can be decoded by reader public key.

The First step cookie exchange step is also known as Oakley key exchange protocol.

1. The Tag *T* selects a pseudo-random number $n1$ and sends it along with the message $\{IDt, (d, c)\}$.

2. The Reader *R* upon receiving the message, acknowledges the message and sends its own cookie $n2$ to *T* .

3. The nextmessage from*T* must contain $n2$, else *T* rejects themessage and the Tag verification request.

Security analysis of the Fix

Had Adversary spoofedTag IP address, Adversary would not get $n2$ back from *R*. Therefore Adversary succeedsonly in having *R* send back an acknowledgement, but not in launching the computationally intensive ECC based operations.Therefore the clogging attack is neglectedby these extra steps. It must be noted, though, that this process does notavoid the clogging attack but only resists it to some extent.This fix will completelywork only if $n1$, and $n2$ are encrypted respectively by *T's* and *R*'s private keys for a secure communication [18].

The 2nd solution to fix the clogging attack that present in moosavi.et.al protocol is to encrypt the credential information of step I10 with tag private key so that the message can't be alter for further steps . And also other side reader can decrypt the entire message by using tag public key.

Chapter 5

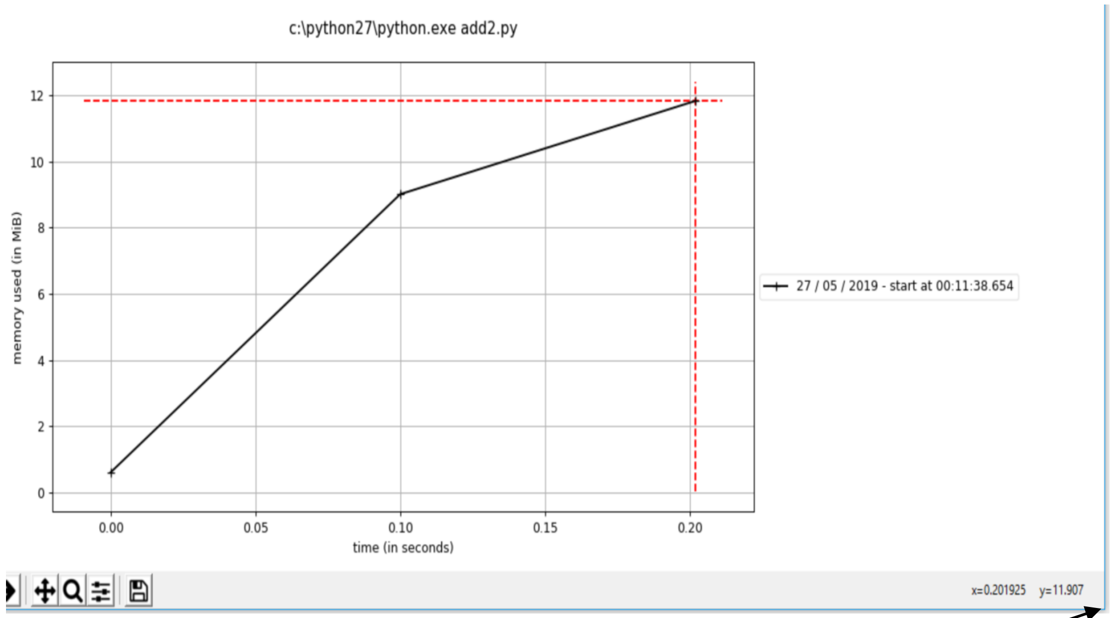
Implementation and Result

5.1 ECC Addition Execution Time

NIST Recommended Curves over prime field	In affine co-ordinates(sec)	In Chudnovsky Co-ordinates(sec)
NIST192$_P$	0.22149	0.21526
NIST224$_P$	0.32975	0.31790
NIST256$_P$	0.43848	0.40897
NIST 384$_P$	0.89617	0.82588
NIST 521$_P$	1.69872	1.50032

Table 5.1 Execution time of ECC addition

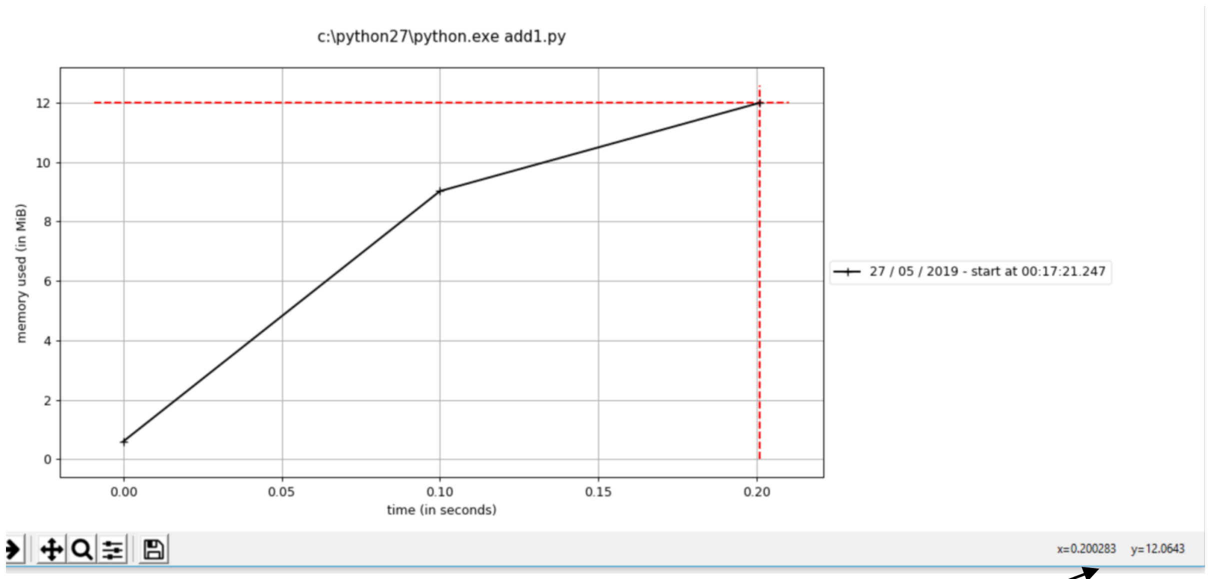
Memory vs time graph of ECC addition in Affine coordinate



X = 0.201925 , Y =11.907

Fig 5.1ECC Addition: memory vs time graph in basic co-ordinates for NIST192$_P$

Memory vs Time graph of ECC addition based on Choudknov Jacobian co-ordinate



X = 0.200283 Sec. and Y = 12.0643 MB

Fig 5.2 ECC addition: Memory vs time graph in suggested co-ordinates for NIST192$_P$

Figure 5.1 and Fig. 5.2 are the memory vs time graphs of ECC addition operation for NIST recommended curve over prime fieldp_{192}. The figure 5.1 is the plot of addition using affine co-ordinates while Fig 5.2 is using Chudnovsky Jacobian co-ordinate system. In Figures 5.1 and 5.2, x- axis indicates time in sec while y-axis indicates memory in MB. From both figures, it is observed that addition using Chudnovsky Jacobian co-ordinate consumes less execution time but extra memory compared to affine co-ordinate. Table 5.1 provides the comparison of ECC addition operation with existing one. Python 2.7.12 based implementation results show that ECC addition operation is faster than existing one.

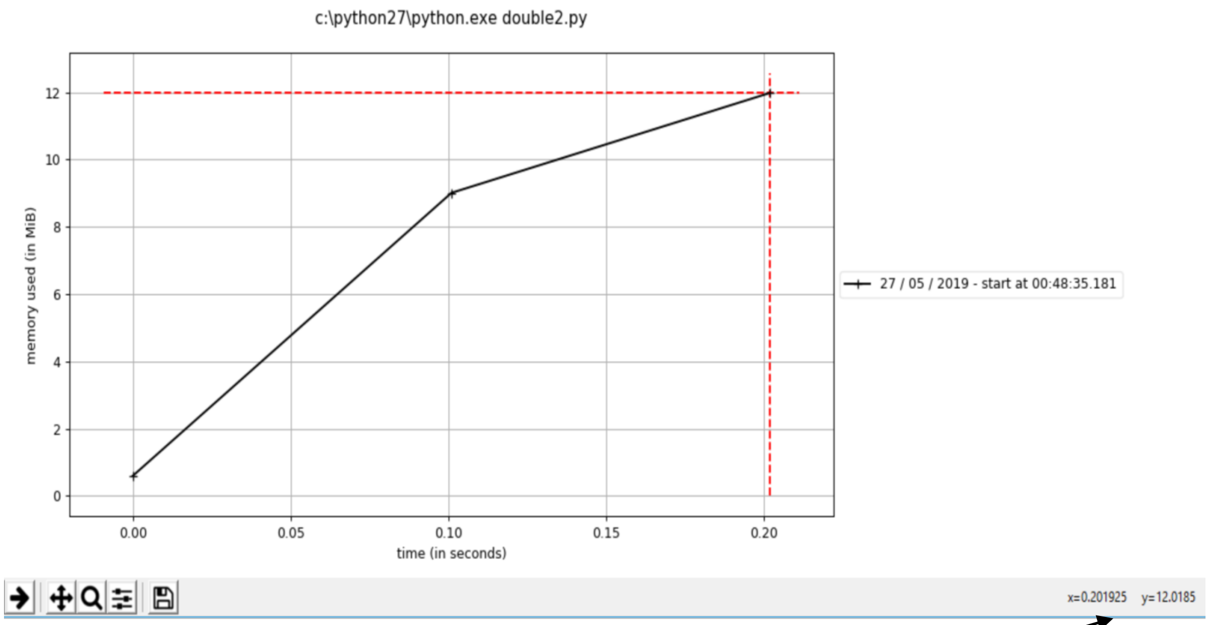
5.2 ECC Doubling Execution Time

NIST Recommended Curves Over prime field	**In affine co-ordinates(sec)**	**In modified Jacobian Co-ordinates(sec)**
$NIST192_p$	0.30284	0.27356
$NIST224_P$	0.32464	0.30415
$NIST256_P$	0.40086	0.38916
$NIST\ 384_P$	1.0534	0.92578
$NIST\ 521_P$	1.8522	1.7414

Table 5.2 Execution time of ECC doubling operation

Table 5.2 is comparison of Execution time analysis of ECC doubling operation in affine co-ordinate and modified Jacobian co-ordinate. We achieve the efficient computation time.

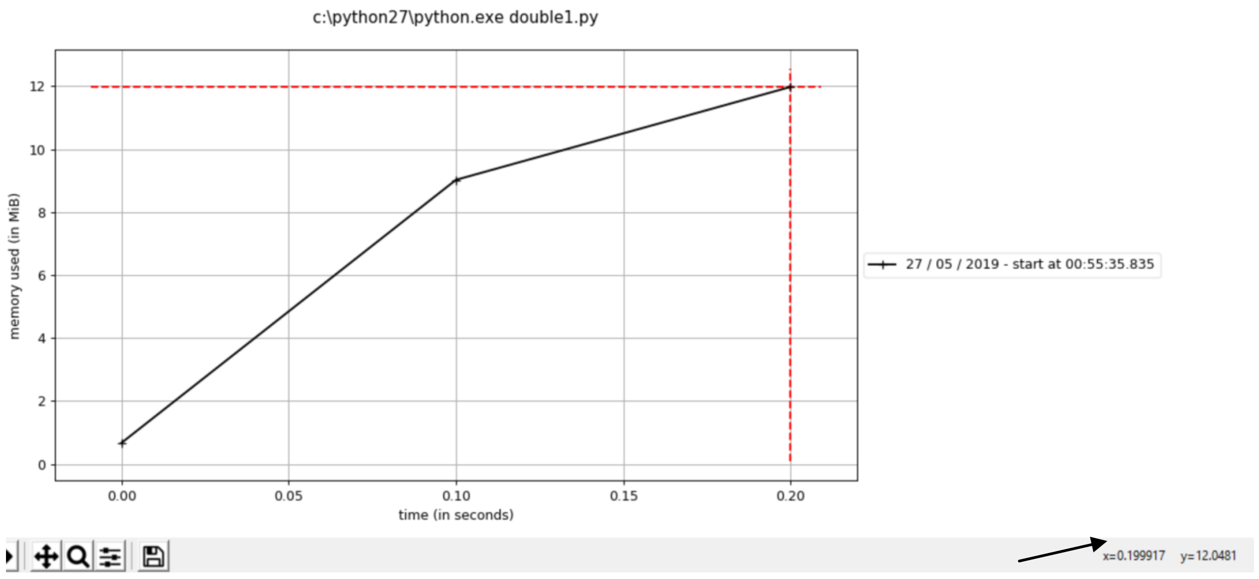
Time vs Memory graph for doubling in affine co-ordinates



X = 0.201925 sec, Y =12.0185 MB

Fig 5.3: Time vs Memory graph for doubling in basic co-ordinates for $NIST192_P$

Time vs Memory graph for doubling in modified Jacobian co-ordinates



X = 0.199917 Sec,Y = 12.0481 MB

Fig 5.4: Time vs memory graph for doubling in suggested co-ordinates for NIST192$_P$

Figure 5.1 and Fig. 5.2 are the memory vs time graphs of ECC doubling operation for NIST recommended curve over prime fieldp_{192}. The figure 5.4 is the plot of doubling using affine co-ordinates while Fig 5.2 is using modified Jacobian co-ordinate system. From both figures, it is observed that doubling using modified Jacobian co-ordinate consumes less execution time but extra memory compared to affine co-ordinate.

5.3 ECC Scalar Multiplication Execution Time

NIST Recommended Curves Over prime field	**Suggested Algorithm in Affine co-ordinates**	**Suggested Algorithm based on Inversion free co-ordinate**	**In traditional method**
NIST192$_p$	0.31277	0.22249	0.34189
NIST224$_P$	0.35912	0.32968	0.36094
NIST256$_P$	0.45277	0.41975	0.47167
NIST 384	0.98254	0.90236	1.0195
NIST 521	1.7956	1.7034	1.8596

Table5.3: Execution Time of ECC scalar Multiplication

Memory vs Time Graph for Scalar Multiplication in Affine Co-ordinates

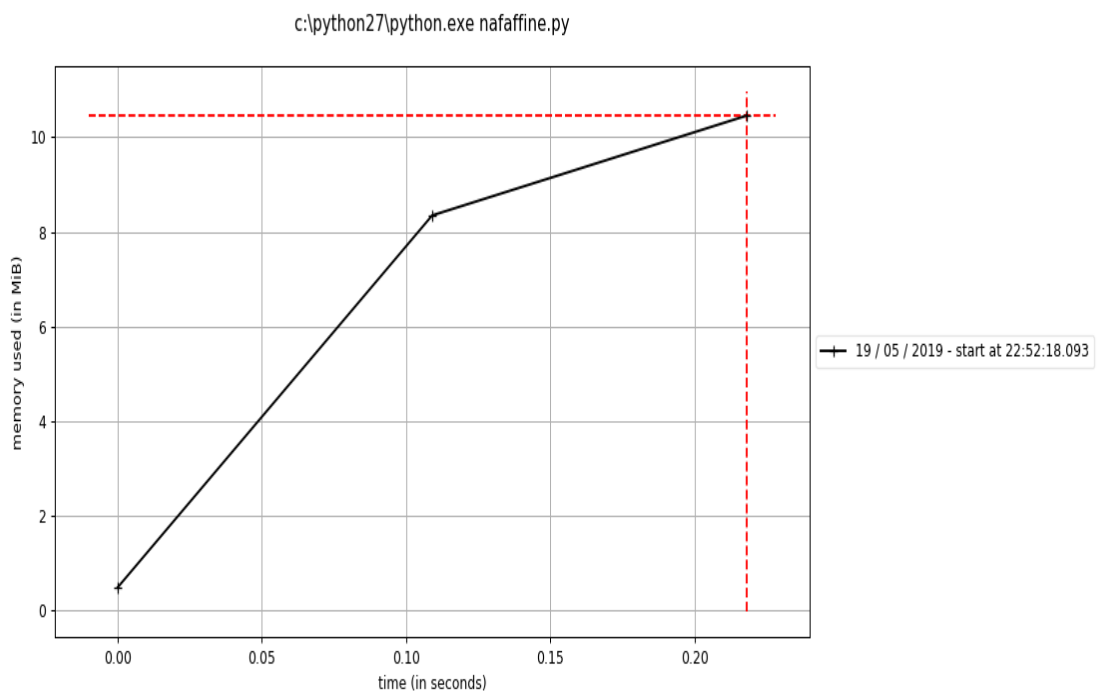


Fig 5.5 Time vs Memory graph of scalar multiplicationin Affine co-ordinate for NIST192p

Memory vs Time graph for scalar multiplication in Inversion free co-ordinate

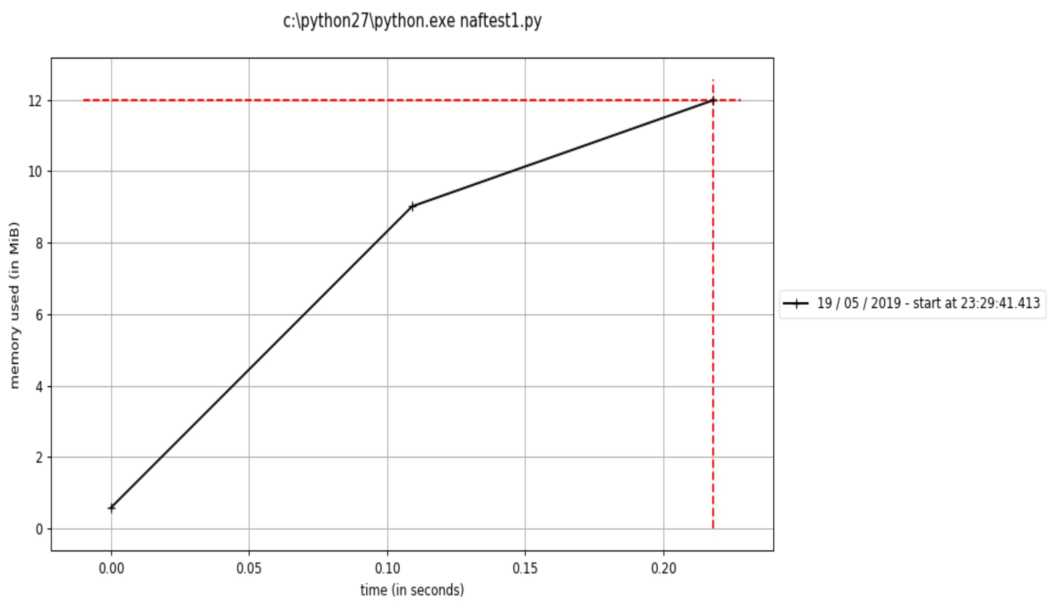


Fig 5.6 Time vs Memory graph of scalar multiplicationfor NIST192p

From fig 5.5 and fig 5.6 of Memory vs time graph in different co-ordinates, we observed that inversion free co-ordinate system is computationally efficient while consumes extra memory. Similarly in affine co-ordinate we observed that it is computationally inefficient while consumes less memory.

5.4 Elliptic Curve Digital Signature Generation Time

NIST Recommended Curves Over prime field	With the use of traditional scalar multiplication	With the use of proposed scalar multiplication
NIST192$_p$	0.16135	0.12022
NIST224$_P$	0.22096	0.14940
NIST256$_P$	0.47038	0.23168
NIST 384$_P$	0.70824	0.42579
NIST 521$_P$	1.5029	0.88714

Table 5.4 Execution Time of ECC based Signature generation Scheme

5.5 ECDSA Execution Time

NIST Recommended Curves Over prime field	With the use of Traditional scalar multiplication	With the use of Suggested scalar multiplication
NIST192$_p$	0.18102	0.14102
NIST224$_P$	0.25142	0.18181
NIST256$_P$	0.34245	0.26588
NIST 384$_P$	0.75025	0.60950
NIST 521$_P$	1.5212	1.1901

Table 5.5 Execution Time of ECDSA scheme

5.6 Summary:

One of the challenging issue in implementing light weight scalar multiplication for embedded in light weight IoT protocols. In our thesis we implemented a light weight ECC scalar multiplication and verified the result in Python 2.7.12.The selection of best suited co-ordinates system was also a challenging work, because we know that in basic co-ordinates an inversion operation is mandatory for doubling and addition operation which is costly. We have implemented the scalar multiplication in both co-ordinates. And we got a time vs. memory graph of proposed scalar multiplication in fig 5.5 and 5.6. We have selected a NIST recommended curves over prime field for all results. We have used different co-ordinates for addition and doubling operation. Because in a fix co-ordinate one operation is faster while another is slower. In section 5.4 we have seen signature generation execution time and in section5.5 ECDSAexecution time with embedded proposed scalar multiplicationand compared with existing one.

There are multiple figure of Time vs Memory in this chapter 5. In case of ECC addition operation the figure verified that in the suggested operation takes some extra memory requirement. Similarly In the case of doubling operation. So finally according to system requirements we can choose proper operation.

All results have been checked by Python programming language with HP pavilion laptop, processor AMD8, and 6GBRAMareits specification.

Chapter 6

Conclusion and suggestion for future work

6.1 Conclusion

In our model we have used both types of co-ordinates to improve the ECC-based arithmetic operation. Our main aim was to improve scalar multiplication with limited available memory. We have improved the ECC scalar multiplication computational time in two cases. In first case with limited memory, we have improved the computational time and in second case we have further improved the computational time at the cost of extra memory. We have embedded this scalar multiplication in ECDSA. We observed that in ECDSA multiple time scalar multiplications are used. So a significant improvement in signature generation and verification is noticed. We also analyzed an ECC based protocol where scalar multiplications are used in many times. If proposed scalar multiplication is embedded in an ECC based protocol then we can achieve significant improvement.

In this thesis clogging attack is demonstrated and it is used to analyse Moosavi et al. protocol and also two different counter measures are suggested. ECC based scheme provides a high level of security. Hence ECC based authentication and key exchange protocols are popularly used to provide security in smart card or RFID.

In chapter 5, timing results of scalar multiplication and ECDSA for all NIST recommended curves are presented. All ECC arithmetic operations have been tested using Python 2.7.12. Timing results are compared with the traditional ECC arithmetic operations and approximately 8-12% improvement has been achieved.

6.2 Future Scope:

There are several areas of future work and some of them are as follows.

1. The software or hardware implementation of suggested scalar multiplication on microprocessor or DSP

2. An efficient signature generation scheme can be used in any ECC based protocol for authentication.

3. An extensive study on different NIST-recommended curves is necessary for implementing a Security protocol for a resource constrained device such as RFID.

Research on ECC based authentication protocol is ongoing. A research on ECC based RFID authentication protocol has been done by static analysis. Obviously next step will be to check dynamic vulnerabilities of this protocol.

References:

[1] I. Riedel, “Security in Ad-hoc Networks: Protocols and Elliptic Curve Cryptography, on an Embedded Platform.” Ruhr- University at Bochum, March 2003.

[2]G.Stoneburner, Underlying technical models for information technology security. Recommendations of the National Institute of Standards and Technology, NIST Special Publication 800-33, 2001

[3] A. J. Menezes, Elliptic Curve Public Key Cryptosystems. Boston: K luwer Academic Publishers, 1993

[4] W. Stalling, Cryptography and network security principal and practices fourth edition

[5] E. Tata “Elliptic Curve Cryptography, An Implementation Guide.” In *Anoop MS*. India: Anoop, MS, 2007

[6] M. Brown, D. Hankerson, J. Lopez and A. Menezes, “Software Implementation of the NIST elliptic curves over prime fields,” in *Progress in Cryptology CT-RSA 2001* , Vol. 2020 of Lecture Notes in Computer Science, pp. 250-265, Springer-Verlag, 2001.

[7] D. Hankerson, A. Menezes and S. Vanstone, “Guide to Elliptic Curve Cryptography” Springer-Verlag, 2004.

[8] R. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen and F. Vercauteren, “Handbook of Elliptic and Hyperelliptic Curve Cryptography,” CRC Press, 2005

[9] O. Billet and M. Joye, “Fast Point Multiplication on Elliptic Curves through Isogenies,” Applied Algebra, Algebraic Algorithms and Error-CorrectingCodes, LNCS Vol. 2643, pp. 43–50, Springer-Verlag, 2003

[10] H. cohen, A. Miyaji and T.ono, “Efficient Elliptic Curve ExponentiationUsing Mixed CoordinatesMultimedia Development Center, Matsushita Electric IndustrialCo., Ltd. 2003

[11] P. C. vanOorschot and J. M. Wiener Parallel Collision Search with CryptanalyticApplications,1996

[12]J. Silverman and J. Stapleton. Contribution of the ANSI X9F1 working group,1997

[13] **A.** Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field", IEEE Transactions on Information Theory, vol. 39, pp. 1639-1646, 1993

[14] A. Kayali, M. A. Khaled, "Elliptic Curve Cryptography and Smart Cards." SANS Institute, 17 February 2004: 2004.

[15]J. Jesper,The Most Misunderstood Windows Security Setting of All Time.TechNet Magazine.Retrieved on 2007-01-08., 2006.

[16]A. Menezes, P. V.Oorschot, and S. Vanstone,*Handbook of Applied Cryptography.* CRC Press, 1996

[17] K. Garrett, S. R. Talluri, and S. Roy, (2015). On vulnerability analysis of several password Authentication protocols. Innovations in Systems and Software Engineering, 11(3):167–176

[18] C. Khatwani and S. Roy,(2015). Security analysis of ecc based authentication protocols. In Computational Intelligence and Communication Networks (CICN), 2015 Interna-tional Conference on, pages 1167–1172. IEEE

[19] L. Xu, and F. Wu, (2015). An improved and provable remote user authentication scheme based on elliptic curve cryptosystemwith user anonymity. Security and Communica-tion Networks, 8(2):245–260.

[20] M. S. Farash, and M. Ahmadian-Attari, (2014). A pairing-free id-based key agreement protocol with different pkgs. IJ Network Security, 16(2): pp143–148.

[21]J.K.Higgins, (2008). Hacker's Choice: Top Six Database Attacks. http://www.darkreading.com/risk/hackers-choice-top-six-database-attacks/d/d-id/1129481?/.

[22] FIPS 186-2, "Digital signature standard (DSS)", National Institute of Standards and Technology, 2000

[23] H. Cohen, A. Miyaji, and T. Ono., "Efficient elliptic curve exponentiation using mixed coordinates", Advances in Cryptology, Springer- Verlag Lecture Notes in Computer Science vol. 1514, pp. 51-65, 1998

[24] I. Semaev, "Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p", Mathematics of Computation, vol. 67, pp. 353-356, 1998

[25] N. Smart, "The discrete logarithm problem on elliptic curves of trace one", Journal of Cryptology, vol. 12, pp. 193-196, 1999

[26]]T. Satoh and K. Araki, "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves", CommentariiMathematiciUniversitatis Sancti Pauli, vol. 47, pp. 8 1-92, 1998

[27] K. Finkenzeller, (2003): "RFID Handbook: Fundamentals and Applications in Contactless SmartCards and Identification", Second Edition, John Wiley & Sons, Ltd. ISBN 0-470-84402-7

[28] Z. Asif, and M. Mandviwalla (2005) "Integrating the Supply Chain with RFID: A Technical and Business Analysis," Communications of the Association for Information Systems: Vol. 15 , Article 24,2005

[29]R. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen and F. Vercauteren, "Handbookof Elliptic and Hyperelliptic Curve Cryptography, "CRC Press, 2005

[30] R. Avanzi, V. Dimitrov, C. Doche and F. Sica, "Extending Scalar Multiplication Using Double Bases," Proc. *AsiaCrypt'06*, LNCS Vol. 4284,pp. 130-144, 2006

[31] K. Aoki, F. Hoshino, T. Kobayashi and H. Oguro, "Elliptic Curve Arithmetic Using SIMD," *ISC2001*, Vol. 2200 of Lecture Notes in Computer Science, pp.235-247, Springer-Verlag, 2001

[32]G.Avoine, E. Dysli, and P. Oechslin. "Reducing time complexity in RFID systems."*In B Preneel and S. Tavares, editors, Selected Areas in Cryptography – SAC 2005,*.Lecture Notes in Computer Science. Springer-Verlag., 2005.

[33] L.Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede. *An elliptic curve processor suitable for RFID-tags.*Report 2006/227: Cryptology ePrint Archive, 2006

[34]S.Bellovin, and E. Rescorla. "Deploying a New Hash Algorithm."*In a presentation delivered at the Rump Session of CRYPTO 2005*, 2005

[35] D. V. Chudnovsky and G.V. Chudnovsky, "Sequences of numbers generated by addition in formal groups and new primality and factorization tests", Advances in Applied Mathematics, VO~.7, pp. 385-434, 1987.

[36] R. Balasubramanian and N. Koblitz, "The improbability that an elliptic curve has Subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm", Journal of Cryptology, vol. 1 1, pp. 1 14-145, 1998.

[37] B. A.Gordon, R. C. Mullin, and S. A. Vanstone. An implementation of elliptic curve cryptosystems over . IEEEJournal on Selected Areas in Communications, 11(5):804–813, 1993

[38] I. F. Blake, G. Seroussi, and N. Smart.Elliptic Curves in Cryptography. London Mathematical Society Lecture Note Series 265.Cambridge University Press, Cambridge, 1999

[39] B. G. Agnew, C. R. Mullin and S.A. Vanstone. An implementation of elliptic curve cryptosystems over . IEEE Journal on Selected Areas in Communications, 11(5):804–813, 1993.

[40] H. Cohen, A. Miyaji, and T. Ono. Efficient elliptic curve exponentiation using mixed coordinates. In Advances inCryptology-ASIACRYPTO'98, LNCS 1514, pages 51–65. Springer-Verlag, 1998.

[41] T. ElGamal, A public key cryptosystemand a signature scheme based on discrete logarithms. IEEE Transactions on InformationTheory, IT-31(4):469–472, 1985.

[42] G. Locke and P. Gallagher, Fips pub 186-3:Digital signature standard (dss). federal information processing standards publication.National Institute of Standards andTechnology, 2009.

[43]SR. Singh, AK. Khan, Performance evaluation of RSA and Elliptic Curve Cryptography. In Contemporary Computing and Informatics (IC3I), 2016 2nd International Conference on 2016 Dec 14 (pp. 302-306). IEEE.

[44] S. R. Moosavi, E. Nigussie, S. Virtanen and J. Isoaho, (2014). An elliptic curve-based mutual authentication scheme for rfid implant systems. *Procedia Computer Science*, 32:198–206.

[45] S. V. Miller, Use of elliptic curves incryptography. In Advances in Cryptology-CRYPTO'85 Proceedings (C. S. Barbara, 1985), volume 218, pages 417–426.Springer-Verlag, 1986

[46] I. Koyama and Y. Tsuruoka, "Speeding up elliptic cryptosystems by using a signed binary window method", Springer-Verlag Lecture Notes in Computer Science, vol. 740, 1993

[47] T. Hasegawa, J. Nakajima and M. Matsui, "A small and fast software implementation of elliptic curve cryptosystems over *GF(p)* on a 16-bit microcomputer", IEICE Trans. Fundamentals, vol. E82-A, no. 1, 1999.

[48] D. Johnson, A. Menezes, S. Vanstone, “The Elliptic Curve Digital Signature Algorithm (ECDSA).

[49] SK. Verma and DB.Ojha, A discussion on elliptic curve cryptography and its applications. IJCS International Journal of Computer Science. 2012(2012)

[50] R.S. Katti, "Speeding up elliptic cryptosystems using a new signed binary representation for integers", Proc. Euro-Micro Conf. Digital Design, 2002

[51] M. Rivain, “Fast and Regular Algorithms for Scalar Multiplication over Elliptic Curves” IACR Cryptology ePrint Archive 2011, p.338, 2011

[52] A. Venelliand F. Dassance, “Faster Side- Channel Resistant Elliptic Curve Scalar Multiplication,” Arithmetic, Geometry, Cryptography and Coding Theory 2009, volume 521 of Contemporary Mathematics, pages 29–40. American Mathematical Society, 2010

[53] P. C. Kocher, “ Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and other Systems,” Advances in Cryptology – CRYPTO ’96 vol. 1109 of Lecture Notes in Computer Science, pp. 104–113, 1996

[54]] P. Longa and C. Gebotys, “Efficient Techniques for High Speed Elliptic Curve Cryptography,” Cryptographic hardware and embedded systems, CHES 2010, p80-94, 2010

[55] D. McGrew, K. Igoe, M. Salter, RFC 6490, “Fundamental Elliptic Curve Cryptography Algorithms,” February 2011, http://tools.ietf.org/html/rfc6090

[56] P. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. Mathematics of Computation, volume 48, pages 243–264, 1987