ELLIPTIC CURVE CRYPTOGRAPHY ON AN IMAGE

Thesis Submitted in Partial Fulfillment of the Requirement for the Award of the Degree of

Master of Technology

IN

INFORMATION SYSTEMS

SUBMITTED BY

MADHU CHOUDHARY

(2K11/ISY/13)

UNDER THE GUIDANCE OF

Mr. N.S. RAGHAVA

ASSOCIATE PROFESSOR

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

BAWANA ROAD, DELHI-110042

(2011-2013)

CERTIFICATE

This is to certify that the thesis entitled **"Elliptic Curve Cryptography On Image "** submitted by **Madhu Choudhary (2K11/ISY/13)** to the Delhi Technological University, New Delhi for the award of **Master of Technology** is a bonafide record of research work carried out by her under the supervision.

The content of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

N.S.Raghava

Associate professor

Delhi Technological University

Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I express my gratitude to my project guide, **N.S. Raghava, Associate professor Department of Information Technology, Delhi Technological University**. He provided a motivation and enthusiastic atmosphere during the many discussions we had. He has changed my thought process radically with his critical approach to problem. Despite his hectic schedule he was always approachable and took his time off to attend to my problems and give appropriate advice. Working with him was, indeed, a great learning experience! He will always be constant source of inspiration for me.

It is an immense pleasure to express my sincere gratitude towards my **Prof. O.P.Verma Head of Department of Information, Technology, Delhi Technological University**, for this immaculate guidance. I consider myself extremely fortunate to have a chance to work under his supervision. It has been very enlightening and enjoyable experience to work under them.

I humbly extend my words of gratitude to other faculty members of this department for providing their valuable help and time whenever it was required.

Place:

Date:

Madhu Choudhary

M Tech ISY

2k11/isy/13

ABSTRACT

Information Security is an area which is concerned about secured transferring of data over public networks. The technologies usually used for providing security require more number of bits, memory and power, and also requires more time to perform operations. Elliptic Curve Cryptography (ECC) is used to overcome these disadvantages and further improve the characteristics of the digital image.

Data encryption is widely used to ensure security in open networks such as the internet. With the fast development of cryptography research and computer technology, the capabilities of cryptosystems such as of RSA and Diffie -Hellman are inadequate due to the requirement of large number of bits. The cryptosystem based on Elliptic Curve Cryptography (ECC) is becoming the recent trend of public key cryptography. This paper presents the implementation of ECC by first transforming the message into an affine point on the Elliptic Curve (EC), over the finite field GF(p). In ECC we normally start with an affine point called Pm(x,y) which lies on the elliptic curve.

In recent years, Elliptic Curve Cryptography (ECC) has attracted the attention of researchers and product developers due to its robust mathematical structure and highest security compared to other existing algorithms like RSA (Rivest Adleman and Shameer Public key Algorithm). It is found to give an increased security compared to RSA for the same key-size or same security as RSA with less key size.

An elliptic curve over a finite field F_p is defined by the parameters a, b $\in$ F_p where a, b satisfy the relation $4a^3+27b^2 \neq 0$, consists of the set of points (x, y) $\in$ F_p that satisfying the equation

$Y^2 \bmod p = X^3+aX+b \bmod p$

The set of points on E (F_p) also include point O, which is the point at infinity and which is the identity element under addition [17].

Each values of a, b give a different elliptic curves. ECC is a public key cryptography technique so it needs two keys one for encryption and another for decryption these keys are called public and private key respectively. Private Key is any random number and public key is a point lie on elliptic curve that is obtained by multiplying the private key with the generator point G in the curve. The generator point G, and the curve parameters ‘a’ and ‘b’ together with few more constants constitutes the domain parameter of ECC.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DLP	Discrete Logarithm Problem
ECC	Elliptic Curve Cryptography
ECDLP	Elliptic Curve Discrete Logarithm Problem
EC	Elliptic Curve
PKI	Public Key Infrastructure
PKC	Public Key Cryptography
RSA	Rivest Shamir Adleman Algorithm
DSA	Digital Signature Algorithm
SEC	Security
PGP	Pretty Good Privacy
PDA	Personal Digital Assistant
RNG	Random Number Generator
AES	Advanced Encryption Standard
DHP	Diffie Hellman Problem

CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Overview:

In the ever expanding digital world, cryptography is becoming extremely important to provide services such as encryption, decryption, key establishment and digital signature to make data more secure. Cryptography is applied into those places where main need of user is security e.g. when a communication takes place between two users in a insecure network and any other user called adversary can easily obtain data from network then security is needed so that adversary is not able to obtain data from communication.

As an another example, user is more concern about credit cards information while doing online transaction so that no one can steal their confidential information, and need more security while sending mail so that no one could try to steal the message, alter selected portions, or pretend to be the user by sending his/her own messages to another user.

It should be evident from these examples that a communicating entity is not necessarily a human, but could be a computer, smart card, or software module acting on behalf of an individual or an organization such as a store or a bank [13]. So cryptography helps to secure data from being viewed or altered and offers a secure communication over insecure channel. To achieve confidentiality, data is encrypted by using cryptographic algorithms and data signature ensures authentication, non-repudiation, and data integrity of the origin of information. If data is encrypted using cryptographic algorithms, transmitted in an encrypted state and decrypted by the intended party then it is difficult to decrypt the encrypted data by third party who intercepts in network channel.

Cryptography is broadly categories into two category public key and private key cryptography. Those categories are based on discrete logarithm problem, factoring problem, and elliptic curve. The use of elliptic curves in public key cryptography was first proposed by Koblitz and Miller in 1980's. The methodologies based on discrete logarithm problem have sub-exponential complexity like sieve method based on discrete logarithm problem using a general number in F_p^* can be solved in sub-exponential time [10]. Whereas a discrete logarithm on an elliptic curve $E(F_p^*)$ has exponential complexity in the size $n= \log_2 q$ of the field element.

1.2 Associated Problems:

Cryptography plays a major role in modern era. It is used in every field such as in any organization, in military, in research, and also used for certificates, for PGP, for internet security, for transport layer security, for blind signature, for secret sharing, and for digital elections etc. There are so many algorithms used in cryptography that fall into a few broad classes such as: inherent key, secret method, hidden key, secret key, public key, and private key. There are many problems or issues associated with those classes that are described below:

When a user uses secret method then he is not aware of dangers that is easy to perform decryption function to decrypt the message because encryption function of secret method is both easy to implement and easy to use,

If data, is encrypted using hidden key cryptography, can be read or viewed using encryption software on your computer by someone who can have access to your computer, or to any other computer on the local network. They don't need any keys or special skill to do this, because the encryption software already has the built in ability to perform cryptanalysis.

There is a key sharing problem in private key cryptography because encryption and decryption function both uses same key. Large key size is needed during encryption process and same as decryption process.

Public key cryptography also has some drawbacks.

First, it is very slow when raising a large number to a large power modulo, than a computation performed on large number required a lot of time, even with the most complicated techniques.

Second, if an attacker able to obtain a person's private key, then it becomes easy to read entire messages that break security.

1.3 Motivation:

Security in digital world and in embedded system has need of an option of an appropriate implementation platform. Elliptic curve cryptography can solve problems in exponential time so it provides an efficient algorithm for finite field F_p^*. It offers better security with smaller key sizes and computationally more efficient algorithm compared to traditional public key cryptosystems such as RSA [10] and Diffie hellman algorithm.

For instance, the elliptic curve digital signature algorithm requires efficient addition, multiplication and inversion in finite field of size larger than 2^{160} [10]. This poses significant problem in embedded systems where computational power is quite restricted and public-key operations are time consuming [10]. A scalar point multiplication operation is performed using a subsequence of huge number of field multiplications and inversions in elliptic curve. To perform an inversion method becomes more complex and many times more costly as compare to multiplication. One of the possible solutions with the above problem is to use projective coordinates for indicating the points on the curve. But this solution is not sufficient because projective coordinates require considerably more temporary storage that is a drawback of it. Affine coordinates are used at the place of projective coordinates so reduces

above problem. Projective coordinates is not essential if the complication of inversion can be reduced considerably. So in this thesis, an issue that is discussed above is reduced by proposing a new method that helps to perform inversion method easily.

1.4 Problem Definition:

Operation that plays major role in public key cryptography is finite field arithmetic operation in which an operation is performed on limited numbers that exists within finite field and result is also closed in same field. But the operation performed in RSA, Diffie Hellman key exchange and DSA is modular exponentiation, which can be performed by decomposing into many modular squaring and modular multiplication. Even though modular squaring operation is performed comparatively more efficient than modular multiplication, both operations follow same procedure i.e. a modular multiplier.

Cryptosystems based on elliptic curve perform point multiplication operation which consists of succeeding point doublings and point additions/subtractions operations. An efficient way to perform modular multiplication is iterative multiplication method that helps to achieve a substantial speedup compared to the recursive multiplication method. Using the iterative multiplication method, encryption method is performed in exponential time complexity and also reduces required memory spaces to maintain each recursively calling function, their domain parameters, registers, and backtracking. Using this method, multiplication of large numbers can be performed only in finite iterations and no need of extra temporary storage.

Affine points are used in ECC for representing points on curve and are also used to reduce required temporary storage. Inversion method is more complex and several times more costly than multiplication. So in this thesis discrete logarithm problem is used to perform inversion method to reduce cost and make it easy.

1.5 Organization of Thesis:

This chapter start to give a introduction of cryptography, and then describes all techniques that is used in cryptography, discuss all problem that occur in different technique of cryptography that break the security and discuss all facts which motivate to do work on ECC, discuss all the drawback of previously generating techniques.

Remaindering chapters of the thesis are organized as follows.

Chapter 2 starts to give an introduction of cryptography. Then describes all algorithms which are used for implementation of the cryptography, and cryptanalysis and all the methods for cryptanalysis are also introduced.

Chapter 3 describes public key cryptography; apply discrete logarithm problem and factorization problem and comparison between them. Then describe different algorithms for solving problem related to DLP.

Chapter 4 starts to give an introduction of elliptic curves, and describes different methods which represents points and used to perform elliptic curve mathematically. Describe all law of group that applied on curve and Elliptic curve applied on general number, real numbers and finite fields.

Chapter 5 describes elliptic curve cryptography for public-key encryption, digital signatures, and key establishment, generation and validation of domain parameters and key pairs. Describe algorithms which are used to solve Elliptic curve discrete logarithm problem.

Chapter 6 describe all aspects that help to implement elliptic curve cryptography in software and hardware. Examined the generation techniques of keys and also explain key distribution. After that explain encryption function and decryption function and gives a mathematical

explanation of encryption and decryption methods. This chapter also describes the new methodology of multiplication operation that is iterative multiplication method.

Chapter 7 starts to give a software implementation and give complete information of results that is obtained after applied software on an input. This chapter describes how to applied encryption and decryption methods on an image and what output is generated after implementation.

Chapter 8 describes the entire conclusion related to work and discussed about future possible work in this field. This describes how to apply ECC on IPV4 and how efficiently it is working and in future how ECC is helpful to provide security on IPV6.

CHAPTER 2
CRYPTOGRAPHY

2.1 Introduction:

Cryptography is a science in which a message is encrypted and decrypted using mathematics, which means perform transformation of readable form of information into non-readable form and vice versa using mathematical language. Readable form of message is called plain text and the art of transformation of readable form of message into unreadable format is called cipher text. The process of transferring plain text into cipher text is called encryption and the reverse process of encryption is called decryption process. Encryption and decryption process needed keys for transferring one form of data into another form. Before transferring data on network users need to encrypt the message using secret key and that message can be decipher into plain text by the only those who holds a secret key. Cryptanalysis is a technique through which encrypted message sometimes can be broken, that also known as code breaking, although modern cryptography techniques are virtually unbreakable. New form of cryptography came soon after the widespread development of computer communication.

Cryptanalysis is the art and science of analysing information system in order to study the hidden aspect of the system. Cryptanalysis is used to breach cryptographic security systems and gain access to the contents of encrypted message, even if the cryptographic key is unknown. Hidden aspects of system that are used in secure communication are data authentication, integration, confidential, and non-repudiation.

In literature of cryptography taking two users A and B who are communicating to each other and another user C who is adversary and should not able to access the secret information. User A has some confidential information and want to share to user B but not obtained by user C. if user A send information over insecure line then user C can retrieve everything. How can user A keep his information secure? This can be achieved by scrambling the

information in such a way that only user B can obtain information by descrambling it. Scrambling of the message can be performed by encryption schemes. Encryption scheme needs keys that is used on sender side to scramble the message and used on receiver side to descrambling the message to obtain original message.

In this era, the use of internet and communication through electronics become extensive and due to this reason security also have an importance in human beings. So Cryptography is applied due to security purpose which is provided by making data as confidential that not only protect data from alteration and stolen but also used to provide authentication. Corporate data, electronic messages, confidential information and information of credit card, smart card are protected by cryptography. Confidentially is supplied by encryption function that provides a secure communication and protect stored information from disclosure and prevent access of data by unauthorized user. Various cryptographic techniques, including methodology of digital signature and authentication can provide security against spoofing and message forgeries. An essential tool that required to make information secure is cryptography and it is eagerly available on internet to user. There are so many cryptographic systems but 'Pretty Good Privacy' is one of these which are used on internet because it is freely available and effective.

2.2 Various aspects of system:

In above paragraph various aspects of system are considered so these are described below:

2.2.1 Data confidentiality: a simplest way to scramble the data using encryption and decryption so that only sender and receiver can obtain data. Encryption is a conversion of readable form of data into cipher text which helps to provide security and maintain the privacy while sending the data from sender to receiver and through decryption original data can be obtained back form cipher text that is just reverse process of encryption function. The concept of encryption and decryption needs keys. In some situation both encryption and

decryption may need same key while in some situations, encryption and decryption may need different keys.

2.2.2 Authentication: Authentication is a mechanism which ensures that the originator of the message is the one who claimed in the message. This can be made possible by the following process. Suppose, user A sends a message and receiver receives the message but he does not know about originator so he needs a proof which proves the identity of originator that the message is originated by the user A. This can be possible if user A performs an action on message which gives a proof of originator from where message was originated. This is the basic fundamental procedure to check for Authentication.

2.2.3 Integrity: In communication system there is another problem that is the loss of integrity of message being sent by communicating parties. This tells that a message sent by sender can be modified by adversary over communicating path so cryptography process should ensure that the messages sent by sender would not be modified anywhere and receiver receives the original message without any alteration. Cryptographic hash is a methodology which is used to verify the integrity of message

2.2.4 Non Repudiation: there can be a situation where user A sends a message to user B but later on denies that she has actually sent the message. In such situation that is explained above, the originator or sender can be prevented by cryptography to act in such a way. Digital signature is one of the popular method through which this problem can be reduced.

2.3 Cryptanalysis:

As the cryptanalysis discussed above but here a brief introduction is given with the different types of cryptanalysis such as known plain text analysis, chosen plain text analysis, cipher text only analysis, man in middle attack, and timing/ differential power analysis and after discussing types, introducing all basic techniques [2].

2.3.1 Overview: Cryptanalysis refers to the study of ciphers, cipher text, or cryptosystems that is, to secret code systems in order to obtain weaknesses in the system that will allow retrieval of the plaintext from the cipher text, without knowing the key or the algorithm. This process is known as breaking the cipher, cipher text, or cryptosystem. Cryptography is a field of study the complexity and mathematical challenges. It applied on data or message and message or data being an unreadable format so that no one would be able to read the message that was not intended to reader. Before being encrypted a message is known as the plain text. And after applying encryption function it is known to be the cipher text.

Cryptanalysis is a study of cipher text and attempt to bring back the message to plaintext. Cryptanalysis has same level of mathematical challenge and complexity as cryptography. Because of the difficulty concerned with cryptanalysis is only focused on the basic techniques of cryptography needed to encryption mono-alphabetic encryption ciphers and cryptograms.

Cryptanalysis breaks the security that used interchangeably with weakening. It finds the weak points of designing and implementation of the cipher and breaks the security by using brute force attack. Weak points reduce number of keys or attempts required by attack that applies every possible combination of keys until the accurate key s not found. For example, let an implementation uses 128 bits to represent key: this means that a brute force attack would required to attempt all 2^128 possible combinations of key to find the correct key to perform the transformation from cipher text to plaintext. Though, a cryptanalysis discloses a technique that would allow the plaintext to be found in 2^40 rounds. Figure 2.1 is showing how to determine all the strength and weaknesses of the cipher and break the security by calculating the key through all the possible combinations and convert cipher text into plaintext.

Numerous techniques are available for performing cryptanalysis that depends on the access that the cryptanalyst has to be known the plaintext, cipher text, or other aspects of the system. Some of the most common types of cryptanalysis or attack are discussed in section 2.3.2.

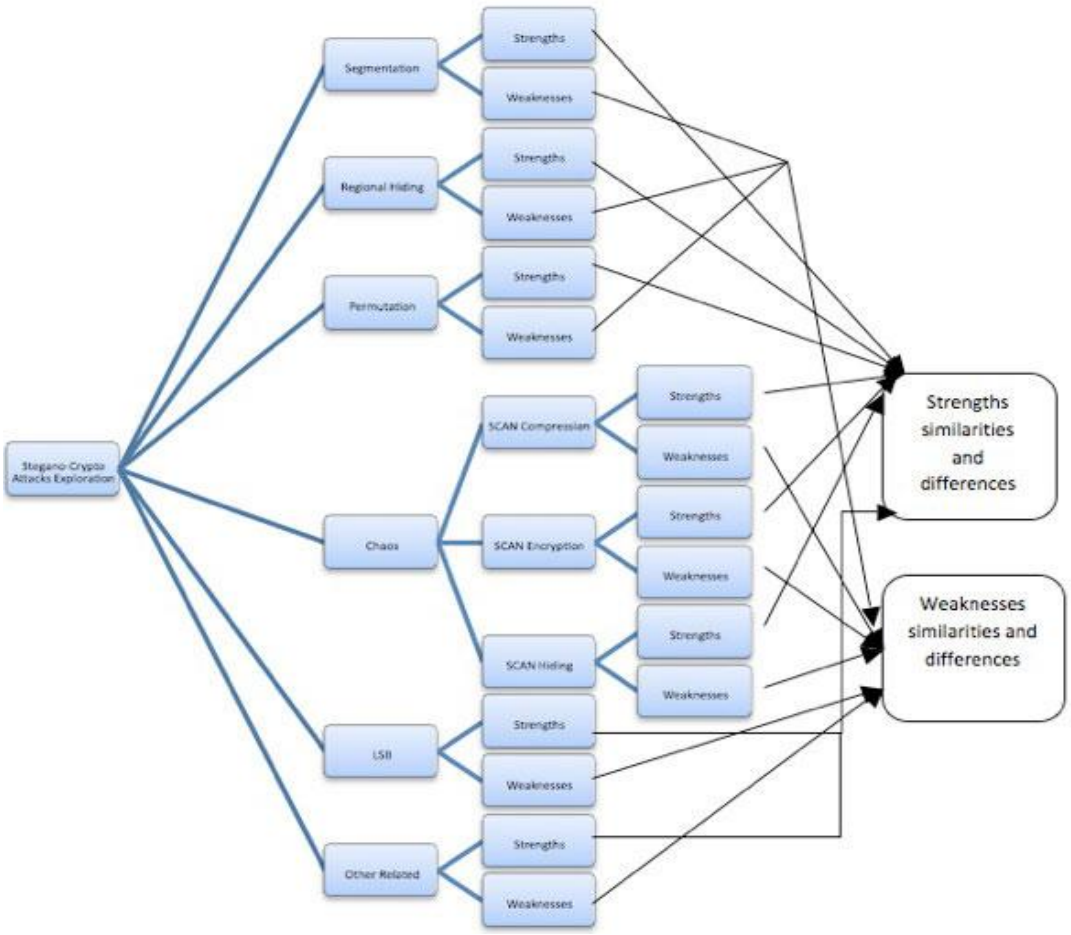

Fig-2.1 Cryptanalysis

2.3.2 Types of cryptanalysis:

Above paragraph gives a brief introduction of cryptanalysis. Cryptanalysis can be performed in many ways depends on situations and known values like if plain text is known then apply in different way, if plain text is not known but cipher text is known then use cryptanalysis in different way etc [14]. Cryptographic attacks are designed to subvert the security of cryptographic algorithms, and they are also applied to decrypt data without knowing the key

and prior knowledge of encryption function. These attacks are part of Cryptanalysis, which is the skill of deciphering encrypted data [26].

2.3.2.1 Known-plaintext analysis: With this procedure, the cryptanalyst has the ability to obtain the portion of the plaintext from the cipher text. Using this information, the cryptanalyst attempts to deduce the key used to produce the cipher text.

2.3.2.2 Chosen-plaintext analysis (also known as differential cryptanalysis): The cryptanalyst is able to have any plaintext encrypted with a key and obtain the resulting cipher text, but the key itself cannot be analyzed. The cryptanalyst is able to deduce the key by evaluating the entire cipher text with the original plaintext. The encryption technique that is somewhat vulnerable to this type of analysis is called The Rivest-Shamir-Adleman technique.

2.3.2.3 Cipher text-only analysis: The cryptanalyst has no knowledge of the plaintext and must work only from the cipher text. This requires accurate guesswork as to how a message could be framed. This analysis helps to have some knowledge of the writing style of the cipher text writer and/or the general subject matter.

2.3.2.4 Man-in-the-middle attack: This differs from the above in that it involves tricking individuals into giving their keys. The cryptanalyst/attacker puts him or herself in the communication channel between two parties who wish to exchange their keys for secure communication (via asymmetric or public key cryptography). The cryptanalyst/attacker then exchanges their keys, with the original parties believing that the keys are being exchanged with each other. The two parties then finally use the keys that are called cryptanalyst/attacker. This type of attack can be overcome by the use of a hash function.

2.3.2.5 Timing/differential power analysis: This is a new technique made public in June 1998, which is mostly useful against the smart card that measures differences in electrical consumption over a time period when a microchip executes a function to make secure

information. This technique helps to expand information about key computations used in the encryption algorithm and other functions related to security. The technique can be applied less effectively by introducing random noise into the computations, or modifying the sequence of the executables to make it harder to monitor the power fluctuations.

2.4 History of Cryptography:

Cryptography is discussed above in 2.1 but this section tells about the origin of cryptography, from where cryptography is started and discusses all the requirements of different techniques, what the reason behind all new methods of cryptography. This section also discussed all techniques on which cryptography is based and used in elderly time these are: Random Number Generation, Primality test, Integer Factorization and Discrete logarithm.

2.4.1 Random Number Generator (RNG)

A random number generator (RNG) is a computational device designed to generate a sequence of numbers that lack any pattern, i.e. appear random. The many applications of randomness have led to the development of several different methods for generating random data [20]. Many of these have continued living from earliest time; including dice, coin flipping, and the shuffling of playing cards, the use of yarrow stalks (by divination) in the I Chin, and many other techniques. Because of the mechanical nature of these techniques in which they generate large amounts of adequately random numbers (important in statistics) required a lot of work and time. Therefore, results would now and then be collected and distributed as random number tables. Nowadays, after the introduction of computational random number generators, it is growing and used in the government-run lotteries, and lottery games, instead of more traditional drawing methods. RNG is also used to conclude the odds of modern slot machines.

Applications of RNGs are statistical sampling, gambling, cryptography, computer simulation, randomized design, and used in other areas where unpredictable results are point of attraction

[20]. By using same key, Sender and receiver can generate the same set of numbers automatically.

RNG plays a vital role in Group Signatures. A basic property of any Group Signature is that it should be untraceable and RNGs help satisfy the property. Each time a member signs a message; randomness in the algorithm ensures that the signatures are different from each other and that no outsider can reveal the identity of the signer from the signature, neither can he claim that two signatures are signed by the same member. RNGs also help reduce the burden of assigning values to parameters required to setup the group.

2.4.2 Primality Test

Primality test is an algorithm used to determine whether an input number is prime. Unlike integer factorization, Primality tests is not able to generate prime factors of a number, it is only stating whether the input number is prime or not. As of 2010, Primality testing is comparatively easy (its running time is polynomial in the size of the input) [44] while factorization is a computationally difficult problem because it helps to generate factors. Some of the Primality tests able to prove that whether a number is prime or not, whereas others prove that a number is composite or not like Miller-Rabin. Consequently call the final compositeness tests instead of Primality tests.

Primality tests appear in two categories: deterministic and probabilistic.

2.4.2.1 Deterministic Algorithm: A deterministic Primality testing algorithm accepts an integer and always outputs a prime or a composite. Deterministic tests determine with absolute certainty whether a number is prime. Until recently, all deterministic algorithms were so insufficient at finding larger primes that they were considered infeasible. In 2002, Agrawal, Kayal and Saxena announced that they had found an algorithm for Primality testing with polynomial time complexity of O ($(\log^{12} n)$).

2.4.2.2 Probabilistic Algorithm: Probabilistic tests can probably falsely identify that a composite number is a prime with small probability (although not vice versa). Though, it is much faster as compare to deterministic tests. Numbers which are approved by a probabilistic prime test are consequently properly referred to as probable primes until their Primality can be established deterministically.

Fermat's Test: The first probabilistic method is discussed in the Fermat Primality test:

If n is a prime, then $a^{n-1} \equiv 1 \pmod{n}$

Note that if number n is prime then it must be hold congruence but it does not mean that if a number holds congruence, then it is prime. The integer can be prime or composite. We can define the following as Fermat's test:

If n is a prime, then $a^{n-1} \equiv 1 \bmod n$

If n is composite, it is possible that $a^{n-1} \equiv 1 \bmod n$

All primes pass the Fermat's test. Composite may also pass the Fermat's test as well. The bit operation complexity of Fermat's test is same as the complexity of an algorithm that calculates the exponentiation.

Square Root Test: In modular arithmetic, The square root of 1 is either +1 or -1 if n is a prime. The square root is +1 or -1 if n is composite, but there may be other roots. This is known as square root Primality test.

If n is a prime, sqrt (1) mod n= +1 or -1

If n is a composite, sqrt (1) mod n= +1 or -1 and possibly other values.

Miller-Rabin Primality Test: The Miller-Rabin Primality test combines the Fermat's test and square root test in a very elegant and efficient way to find a strong pseudo prime (a prime

with a very high probability of being a prime). In Miller Rabin Primality test n-1 is written as the product of an odd number and a power of two.

$n-1 = m* 2^k$

In other words, instead of calculating a^{n-1} (mod n) in one step, we can do it in k+1 steps. The benefit is that the square root test can be performed in each step repeatedly. This process is performed until square root test fails when it failed then stop and declare that n is a composite number. In each step it is assured that the Fermat's test is passed and the square root test is satisfied between all pairs of adjacent steps, if applicable. It is a probabilistic method that helps to find whether a number is prime or not. There exists a proof in which each time a number passes for the Miller-Rabin Primality Test and the probability of a number that is not a prime is 1/4. If the number passes m tests (with m different bases) the probability that it is not a prime is $(1/4)^m$.

2.4.3 Discrete Logarithm

Discrete logarithms are group-theoretic analogues of ordinary logarithms. Exactly, an ordinary logarithm $\log_a$ (b) is a solution of the equation $a^x = b$ over the real or complex numbers. Correspondingly, if g and h are elements of a finite cyclic group G then a solution x of the equation $g^x = h$ is called a discrete logarithm to the base g of h in the group G [4].

In general, let G be a finite cyclic group with n elements. Let b be a generator of G; then every element g of G can be written in the form $g = b^k$ for some integer k. Furthermore, any two such integer's k1 and k2 representing g will be a congruent modulo n. Then consequently a function can be defined as:

$\log_b: G \rightarrow Z_n$

Where Zn denotes the ring of integers modulo n by assigning to each g is a congruence class of k modulo n. This function is a isomorphism group, called the discrete logarithm to base b.

The familiar base change formula for ordinary logarithms remains valid: If c is another generator of G, then equation is given as:

$\log_c (g) = \log_c (b) * \log_b (g)$

No efficient classical algorithm for computing general discrete logarithms $\log_b g$ is known.

2.4.4 Integer Factorization

Integer factorization or prime factorization both are used to break down a composite number into smaller non-trivial divisors, which when multiplied together equals the original integer.

There is no efficient integer factorization algorithm for very large number; an effort concluded in 2009 by several researchers factored a 232-digit number (RSA-768) utilizing hundreds of machines over a period of 2 years. Many areas of mathematics and computer science face this problem, including quantum computing, elliptic curves, and algebraic number theory.

Not all numbers of a given length are uniformly unbreakable to factor. Instances of these problems are semi primes and the product of two prime numbers.

2.5 Types of Cryptography:

Cryptography is categories into three basic techniques that are so important and valuable in modern era.

- Secret key Cryptography
- Public key cryptography
- Hash function

2.5.1 Secret Key Cryptography

In this cryptography single key is used for encryption and decryption of data. Sender and receiver both applies same key to encrypt and decrypt the message. This technique is called symmetric encryption because only single key is used by sender and receiver both so the biggest problem of this technique is key distribution. This is also called private key cryptography.

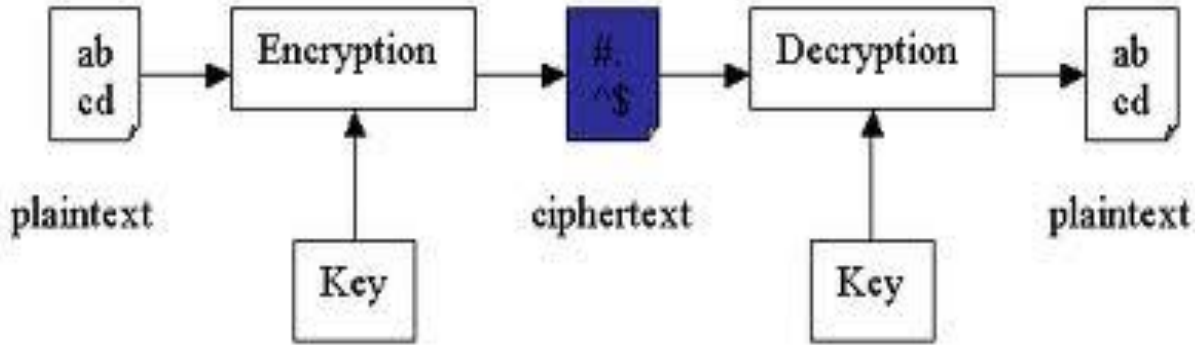

Fig-2.2 secret key cryptography

In figure 2.2 describe the whole procedure of symmetric key cryptography in which a sender using a key and encrypting a message that is ‘abcd’ and send to another user who also use same key for decryption and decrypt the message and gets original message ‘abcd’.

Symmetric-key encryption can use either stream ciphers or block ciphers.[4]

- Stream ciphers encrypt the digits (typically bytes) of a message one at a time.
- Block ciphers take a number of bits and encrypt them as a single packet, and padding the plaintext if required so that it is a multiple of the block size. Size of the commonly used block is 64 bits. The Advanced Encryption Standard (AES) algorithm approved by NIST in December 2001 uses 128-bit blocks [42].

Block cipher based algorithms are AES, DES, Triple DES, CAST-128, IDEA, RC2, RC5 etc. and Stream cipher based algorithm is RC4.

2.5.2 Public Key Cryptography:

Two keys are involved in this type of crypto systems through which a secure communication can be established between communicating user over insecure communication channel. Since two different keys are applied here so this technique is also called asymmetric encryption. A complete procedure of public key cryptography is expressed by figure 2. 3.

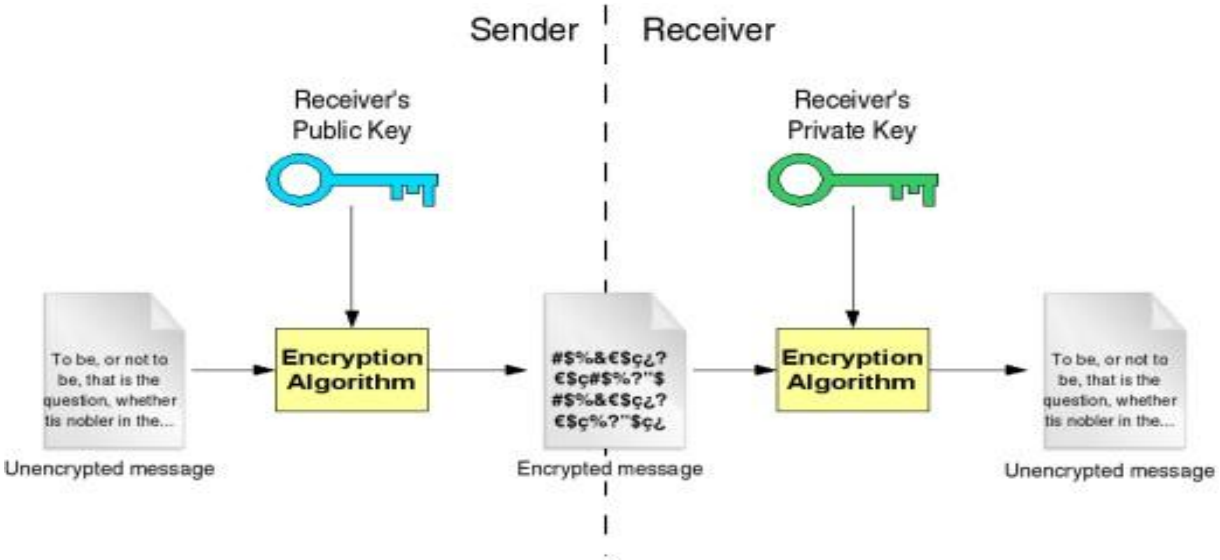

Fig-2.3 public key cryptography

During this procedure, each party generates two keys one of them is private key that is secret key and cannot be disclosed to all and another one is public key which is shared to all communicating users. If user A sends a message to user B, then public key of user B is shared to user A and used to encrypt the message by user A then send over communicating channel and user B uses own private key to decrypt the message.

RSA, Diffie-Hellman Key Exchange, Digital Signature, Elliptic Curve Cryptography etc algorithms are based on public key cryptography. These are used when setup a public key

authentication to login from one server to another server in the backend without having to enter the password.

2.5.3 Hash Functions: In this technique there is no use of any key. However it uses a hash value of fixed length which is determined on the basis of the plain text message. The integrity of the message is ensured by using hash functions and it also keeps a check that the message has not be modified, negotiated or affected by virus.

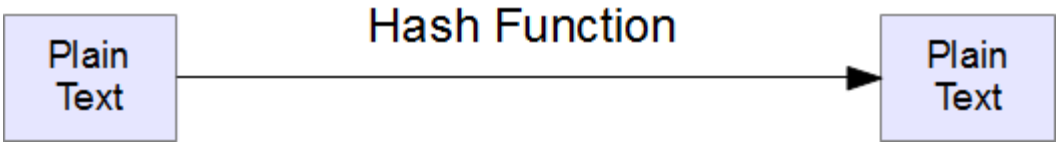

Fig-2.4 Hash function

Functionality of hash function is showing in figure 2.4 and it is clearly that there is no need of any key and any encryption and decryption function, only a hash function is required to generate a hash code of plain text through which the integrity of the message is verified.

2.6 ADVANTAGE AND DISADVANTAGE OF CRYPTOGRAPHY

In cryptography, advantage of an adversary is a measure of how successfully it can attack a cryptographic algorithm, and identifying it uniquely from an idealized edition of that algorithm. Note that in this section, a person is not an "adversary" and an algorithm is acting as an adversary.

Advantages are:

-The messages are hidden by it and maintain privacy.

-by using cryptography a message can be written in any way (any theme any symbol for the

code) to keep it secret.

- Cryptography can be used without knowing the teacher.

Disadvantages of cryptography are:

-It is time-consuming to figure out the code.

-It takes long time to generate the code.

-If you have to send a code to another user in the past, it will take long time to get to that person.

-Overall Cryptography is a lengthy process

CHAPTER 3
PUBLIC KEY CRYPTOGRAPHY

3.1 Encryption Scheme:

In the introduction, we have introduced the notion of an encryption scheme. We now make this idea mathematically precise.

Definition: An encryption scheme is a tuple (P; C; K_E; K_D; E; D), where P, C, K_E and K_D are arbitrary sets (not necessarily distinct), and E and D are sets of functions, such that for each k Є K_E there is a function Ek : P→ C in E, and for every k Є K_D there is a function Dk : C → P in D [23]. This tuple must satisfy the condition that for every t Є K_E, there is a unique s → K_D, such that Ds(Et(p)) = p for all p Є P [23].

The sets P, C, K_E, K_D in the above definition are called the plaintext space, the cipher text space, the encryption key space and the decryption key space respectively. The functions of E and D are encryption and decryption functions respectively.

The main purpose of encrypting message is that someone who intercepts that message does not get knowledge of the original message. Therefore it seems reasonable to demand of encryption schemes so that it become hard to decrypt cipher text by someone who does not have decryption key. So methodology of cryptanalysis like guessing and calculating plaintext should be a difficult task and it should be easy to encrypt and decrypt by someone who knows the keys.

Above some words like ‘difficult’, ‘hard’, and ‘easy’ are used. From now on, with an ‘easy’ problem, mean a problem that can be solved by some algorithm in polynomial time, and a problem is ‘hard’ if it cannot be solved by any algorithm in polynomial time. This means that a hard problem take huge amount of time to solve while an easy problem can be solved relatively fast [6].

If User A and User B want that no one can obtain plaintext from cipher text so they need to follow encryption schemes and need to keep the used encryption scheme secret. So that someone who wants to intercept the message does not know about encryption scheme otherwise he/she can obtain plaintext from cipher text by applying cryptanalysis.

3.2 Public Key Cryptosystem (PKC):

Encryption schemes are explained in previous section 3.1, where it is stated that in the keys used in encryption key and decryption are same in symmetric key. In this case there is no distinguishing between encryption key and decryption key and treats them as a single object.

A big disadvantage of symmetric encryption scheme is sharing of key. Chosen key needs to share in a secure channel and must remain secret at all time. Even if encryption scheme is highly sophisticated and extremely safe, but as soon as the key moves out in the open, security is violated.

The biggest security-threat is the exchange of the key. When User A and User B communicate with each other about the key, there is frequently a chance for Eve to catch this key.

To overcome this disadvantage of symmetric encryption scheme, public key cryptosystem were invented. Public key cryptosystem is based on asymmetric encryption scheme in which a key pair is used one of them is secret key that is used as a decryption key and another is shared to all communicating user that is used as a encryption key. An idea behind a PKC is that User A has a pair of key one of them is public key and another one is private key [3]. She chooses public key which she publishes. Now User B retrieve public key of User A and encrypt the message and send to User A. She chooses private key for herself and keep it secret. Then decrypt

the User B's message with the help of her secret key. To get a secure cryptosystem this way, determining the secret key with the help of public key should be a hard problem.

Public-key cryptography is based on asymmetric key algorithms and also referred by a more generic term "asymmetric key cryptography." The techniques used to conduct public key cryptography are done on the basis of mathematical relationships ones of being the integer factorization and discrete logarithm problems. In public key algorithm there is no need of initial exchange of keys in a secure channel between communicating users as needed in symmetric key algorithms. The authenticity of a message is also checked by the use of these algorithms with the help of a digital signature of the message using the private key, and then the public key is further used for verification. For signature verification purposes, only hash of the message is usually encrypted, so public key distribution and digital signature both operations are performed by public key cryptosystem [3].

3.3 Integer Factorization Problem:

In number theory integer factorization and prime factorization are two things which means decompose a composite number into smaller nontrivial divisor in such a way that their multiplication gives the original number [24]. Factorization means decompose a positive integer into positive integers n1 and n2 such that their product is equal to n (i.e. n1.n2=n), and both n1 and n2 are larger than 1. Such n1 and n2 are called factors (or divisors) of n. if positive integers greater than 1 that cannot be decomposed or factored are called primes.

Problem which is performing factorization of a composite integer is believed to be a hard. Of course, composition of small factors is easy to factor but large factors are hard and problem seems to be difficult. Widely used algorithm in cryptography is

based on the difficulty of factoring problem such as RSA. In RSA factoring modulus would allow an attacker to analyse the private key. Thus, messages can be decrypted and signatures can be forged by anyone who can factor the modulus. RSA algorithm becomes more secure if difficulty of factors is high and the absence of other types of attack. As namely, in RSA both parties choose large primes then take modulus of the product of those primes and hence an attacker requires more time and efforts to factor the larger primes. Thus far, consider that a number consists of large prime factors might hold definite properties making it easy to factor.

There are two different version of problem described as follows:

The function problem version: If an integer n is given then this version determine another integer n1 which satisfy the following constraint $1 < n1 < n$ that divides N [24]. This problem is insignificantly in FNP and it's not known whether it lies in FP or not. Most practical implementations can be used to solve this version.

The decision problem version: If an integer n and another integer m is given with some constraint $1 \leq m \leq n$, does N have a factor d with $1 < d < M$? In this version most well-studied complexity classes are categories as decision problems, not function problems [24]. This version is used frequently for optimization problems, because it can solve a function problem in a logarithm number by applying decision problem version along with binary search.

3.4 Discrete Logarithm Problem (DLP):

In this thesis all public key cryptography‘s are based on the difficulty of the discrete logarithm problem (DLP). This problem is defined as:

Let G be an abelian (additive) group, and g Є G. Now suppose that h Є <g> G. We can ask ourselves which k Є Z satisfies the identity kg = h. Finding such a k is the discrete logarithm problem. More commonly:

Definition Let G be a finite cyclic group containing n elements. It is supposed that the group is written multiplicatively [4]. Let us assume that b be a generator of G then every element g of G can be written in the form $g = b^k$ for some integer k. Furthermore, any two such integer's k_1 and k_2 representing *g* will be congruent modulo *n*. We can thus define a function [4].

$\log_b = G \rightarrow Z_n$

Discrete logarithms are perhaps simplest to understand in the group $(Zp)^{\times}$ [4]. Discrete Exponentiation method used to determine k^{th} power of a number that can be done by determining k^{th} power of integer and then calculate remainder after dividing by p. For example, consider $(Z_{17})^{\times}$. To compute 3^4 in this group, first compute $3^4 = 81$, and then divide 81 by 17, obtaining a remainder of 13. Thus $3^4 = 13$ in the group $(Z17)^{\times}$ [4].

The difficulty of the DLP depends on the underlying group. In this thesis, we will focus on the groups given by elliptic curves. In Section 3.6 we will encounter some general algorithms that will solve the DLP for any group. These algorithms take exponential time and are therefore slow. For the groups of the form F^*_P there exists a slightly faster algorithm, which we discuss in Section 3.6. This algorithm takes sub exponential time.

3.5 Example of public key cryptosystem based on DLP:

In section 3.2 we have discussed notion of a public key cryptosystem. Now we are describing the cryptosystem whose security based on the DLP. There are two cryptosystems that is Diffie Hellman Problem and ElGamal Cryptosystem.

3.5.1 Diffie Hellman:

A protocol that is based on symmetric key cryptosystem because it establishes a common key between two communicating parties is called Diffie hellman. This is

used for a large network of users; there can be complicated and logistic secure distribution of keys. So Diffie and Hellman produced a new methodology in 1976 which reduces the above problem.

Diffie hellman key exchange is a methodology to exchange keys which is implemented within the field of cryptography. If two parties that does not have any prior knowledge of each other but want to communicate then Diffie Hellman key exchange method allows them and establish a connection by sharing their secret key over an insecure channel. This secret key is used for encryption and decryption using symmetric key cipher.

Definition. Public group G_1 and an element a_1 are given where $a_1 \in G_1$ of order n, two parties, says User A and User B follows these steps to establish common key:

(a) User A chooses a random integer $a_1 \in Z_n$, calculates $A=a_1^{a1}$ and sends it to User B

(b) User B chooses a random integer $b_1 \in Z_n$ calculates $B=a_1^{b1}$ and sends it to User A

(c) User A calculates $B^{a\ 1}=a_1^{b1a1}$ and User B calculates $A^{b1}=a_1^{a1b1}$ their common key is

$k=a_1^{a1b1}=a_1^{b1a1}$ [30].

The eavesdropper, who knows G and 'a_1' from the public directory, after intercepting A and B, is then faced with the following problem.

Diffie Hellman problem (DHP) is a mathematical problem. The motivation for this problem is that mathematical operations are used in mostly preferred security systems which are computed fast, but not able to perform reverse or being difficult to reverse [29]. For example, encryption is performed on a message then performs reverse

operation of the encryption is difficult. If the DHP can be solved easily then these systems would be easily broken.

Let G be a group and let $a_1 \in G$. given $A=a_1^{a1}$ and $B=a_1^{b1}$ compute $k=a_1^{a1b1}$

If anyone has solution to solve the discrete logarithm problem, then the Diffie Hellman problem can be solved easily without facing any problem. Consequently it is proving that the two problems are comparably equivalent and these equivalents are established for certain cases. At some places the Diffie Hellman key exchange scheme is securely build the Diffie Hellman problem hard. The Diffie Hellman key exchange scheme is widely used to generate "session keys",

3.5.2 ElGamal Cryptosystem: ElGamal system is one of the public-key cryptosystem which is based on the discrete logarithm problem. It provides both mechanism encryption and signature algorithms. The encryption algorithm is equivalent to the Diffie-Hellman key agreement protocol in nature [31].

A prime p and an integer g are the systems parameter, whose power modulo p generates number of elements, by following the same process as in Diffie-Hellman. User A have two key private and public that are represented by 'a_1' and 'y' respectively, where $y = g^{a1} \pmod p$. Suppose User B want to send a message to User A. Initially a random is generated by user B with $k < p$ then determine

$y_1 = g^k \pmod p$ and $y_2 = m \text{ xor } y^k$,

Where xor refers the bit-wise exclusive-or. A pair (y_1,y_2) is sent to user A from user B. After receiving the cipher text, User A determines $m = (y_1^{a1} \bmod p) \text{ xor } y_2$ [31].

The ElGamal signature is an algorithm where public and private key have the same form.

ElGamal signature needs randomness that is main disadvantage of it which slow down its speed (especially for signalling). During encryption message expansion is required

by a factor of two that is another disadvantage of the ElGamal system [31]. However, if the cryptosystem is used only for exchange of secret keys then this disadvantage is negligible.

3.6 Algorithms for solving DLP:

This Section discussed about some algorithms for solving the discrete logarithm problem. This should also give us an idea of how hard the DLP actually is.

Let G be an Abelian group, and g Є G an element of order n, and finally let h Є <g> G. There is an algorithm to find k with kg = h. In naive algorithm the value of b is raised to higher and higher up to powers k until desired value of g is not found; this is also known as trial multiplication. This algorithm needs linear time to applied on the group G and consequently exponential in the number of digits in the size of the group [25].

The naive algorithm is quite slow, in the sense that it takes exponential time. For big n, it becomes infeasible to run the algorithm. Can we do better than that? For general groups, there exist some faster algorithms. However, these still take exponential time. We will discuss two of them: the baby-step giant-step algorithm, and Pollard's ρ-algorithm.

3.6.1 The baby-step giant-step algorithm: The baby-step giant-step algorithm Again, pick an abelian group G, an element g Є G of order n, and some h in the subgroup generated by g. Let $m = |\sqrt{n}|$. Now execute the following steps:

1. Make a list L1 = {0, g, 2g, ….mg}. If h Є L1, we are done, otherwise go to next step.

2. Make a list L2 = {h, h- mg, h- 2mg,….. h - m^2g}: If 0 Є L2, we're done, otherwise go to next step.

3. Determine x Є L1 ∩ L2.

4. x = ig = h - jmg for some $0 \leq i; j \geq m$, hence h = (i + j m)g.

Theorem 3.1. If h Є <g>, the baby-step giant-step algorithm will solve the DLP in $O(\sqrt{n} . \log (n))$ steps.

Proof. We will need to show that the algorithm indeed gives the discrete logarithm of h to base g in G. Suppose h Є L1, then h = ig for some i, and we have found the discrete logarithm. Suppose 0 Є L2, then h - jmg = 0 for some j, and h=jmg, hence jm is the discrete logarithm.

Now suppose that neither h Є L1 nor 0 Є L2. We have assumed h Є <g>, so the discrete logarithm of h to base g exists. Hence h = kg for some $0 \leq k \leq n$. We can write k = qm + r for some $q \leq m$ and $r < m$ (division with remainder).

Hence h = (qm + r)g, so h - qmg = rg. Now h - qmg Є L2, and rg Є L1, so the intersection is non-empty. The element in the intersection can compute the discrete logarithm, so the algorithm indeed solves the DLP [34].

Now we will determine the running time. For step 3, two lists have to be compared. Using a binary search, this takes in the worst case O(m log(m)) steps. Making the first list takes m multiplications, and the same holds for the second list. In total, 2m multiplications are to be processed. Consequently the algorithm requireds

$O(m \log(m) + 2m) = O(\sqrt{n} . \log (n))$ steps.

The baby-step giant-step algorithm solves the DLP significantly faster than the naive algorithm but it still takes exponential time. Another down side is that this algorithm could require a lot of memory, since two lists of size $\sqrt{n}$ need to be stored.

3.6.2 Pollard ρ-algorithm

Another algorithm for solving the DLP is the Pollard ρ-algorithm. It is slightly faster than the baby-step giant-step method and it needs much less storage. Suppose the identity kg = h holds in some abelian group G. Split the group G in three pair wise

disjoint subsets G1, G2, G3, such that the coming together consequential is G. Define a function f : G → G as follows:

Now pick a random $a_0 \leq n = \text{ord } G(g)$, and set $x0 = a_0g$. This x_0 is the first element of a sequence (x0, x1, x2, …….), defined by the recursive relation $x_{i+1} = f(x_i)$. Every entry of this sequence can be written as a linear combination of g and h, so

$x_i = a_ig + b_ih$. We know a_0, and of course $b_0 = 0$. The a_i's and b_i's can be determined for $i > 0$ by the following recursive relations [33]:

Eventually some entry of the sequence defined will have occurred before. More mathematically, there is some $i \geq 0$, and some $t \geq 1$, such that $x_{i+t} = x_i$. Then by definition of the function f, have $x_{(i+1)+t} = x_{i+1}$, $x_{(i+2)+t} = x_{i+2}$, and so on. In other the words, the sequence makes loops, starting from the smallest m for which xm is repeated. The size of the loop equals the smallest positive t for which $x_{i+t} = x_i$.

The fact that a subsequence repeats itself comes in handy for us. If $x_i = x_{i+t}$, then $a_ig+b_ih = a_{i+t}g + b_{i+t}h$. We can rewrite this to $(a_i - a_{i+t})g = (b_{i+t} - b_i)h$, and by assumption this is equivalent with $(a_i - a_{i+t})g = (b_{i+t} - b_i)kg$. From this concluded

$a_i - a_{i+t} - (b_{i+t} - b_i)k \pmod n$.

Now set $v = a_i - a_{i+t}$ and $w = b_{i+t} - b_i$. We thus have the congruence

$wk \equiv v \pmod n$ (1)

If w is invertible modulo n (or equivalently, if gcd(w, n) = 1), then the discrete logarithm k can be computed by $k \equiv vw^{-1} \pmod{n}$. So suppose $d = \gcd(w, n) \geq 2$.

We can find an integer s satisfying $sw \equiv d \pmod{n}$. Now multiply both sides of

(1) by this s to get

$$dk \equiv sv \pmod{n} \qquad (2)$$

But we know that d|n. Now dk = sv+qn for some q, so d/n implies d|sv. Therefore,

$k = \frac{sv+gn}{d}$ is an integer, and is a solution for congruence (2) for every $0 \leq q \leq d-1$.

And this k also satisfies congruence (1) for some value of q, and this k is a discrete logarithm of h to base g.

Let's summarize the steps have taken to find the discrete logarithm:

1. Define the function f, based on g and h.

2. Pick some a0 and compute x0.

3. If x_i is known, compute $x_{i+1} = f(x_i)$. If x_{i+1} have already occurred before reaching to the next step if not, then again repeat same step.

4. Solve the congruence $a_i - a_{i+t} - (b_{i+t} - b_i)k$ obtained from the previous step.

This algorithm solves the discrete logarithm problem. But in this form, every x_i has to be stored so it still requires a lot of storage. Except it there is a clever solution for this memory problem. Besides the sequence (x0, x1, x2, ….), we make another sequence (y0, y1, y2, …….), where y0 = x0 and $y_{i+1} = f(f(y_i))$.

Hence $y_i = x_{2i}$.

During making sequences discard every x_i and y_i from our memory that do not satisfy $x_i = y_i$ (of course, after computing x_{i+1} and y_{i+1}). So only two group elements have to be stored at all time. Once we've found the desired i, we have $x_i = y_i = x_{2i}$, and we have a repetition in our original sequence.

One can ask if this procedure slows down the process of finding the discrete logarithm. The answer is no. In our original algorithm, the sequence gets into a loop that needs m steps to execute, after that loop itself takes t steps, so it takes m+t steps to find a repetition. In the `improvement', we need to find i for which $x_i = x_{2i}$. This happens if $i \geq m$ and

$i \equiv 2i \pmod t$. The equivalence implies that t divides i. But one of m, m + 1, m + 2, …….m + t - 1 is divisible by t. Hence $x_{2i} = x_i$ for $1 \leq i < t+ m$. Hence our improvement does not slow down the process.

The actual speed of the algorithm depends on chance, since it depends on the random a0, and the (random) partition of G. The expected value of m + t is approximately 1, $25\sqrt{n}$. So it is likely that the Pollard ρ-algorithm takes O($\sqrt{n}$) steps.

3.6.3 The Pohlig-Hellman algorithm

The next algorithm that presented here does not solve the DLP itself. However, it does speed up other algorithms (like the ones presented before) when the order of g is a composite number. But first, gives a method that speeds up an algorithm when the order of g is a power of a prime. So suppose that there is an algorithm that finds k satisfying k~g = h in an abelian group G in O(Sp) steps, where p = ordG(~g) is prime. Here Sp is a function of p. For instances, in Pollard's ρ-algorithm, Sp can be represented in the form of p that is expressed as:

Sp = pp.

Now assume that g Є G has order pe, and trying to find k such that kg = h for some h. It is well-known algorithm that can uniquely write k as

$k = k0 + k1p + \ldots\ldots (k_{e-1})(p^{e-1})$; with $0 \leq xi < p$; (3)

since k < pe. Then determine the coefficients of this expression. Since g has order pe, the element $(p^{e-1})g$ has order p. Then

$(p^{e-1})h = (p^{e-1})kg$

$= p^{e-1}(k0 + k1p + \ldots\ldots (k_{e-1})(p^{e-1}))g$

$= (p^{e-1})k_0g + p_eg(k_1 + \ldots\ldots (k_{e-1})(p^{e-2}))$

$= k_{0(}(p_{e-1})g)$

and the equation $k_{0(}(p^{e-1})g) = (p^{e-1})h$ is a DLP. So it can be solved by our assumed algorithm in O(Sp) steps. In other words, it can find the first coefficient of expression (3).

Next compute

$P^{e-2}h = p^{e-2}kg$

$= p^{e-2}(k_0 + k_1p + \ldots\ldots (k_{e-1})(p^{e-1}))g$

$= (p^{e-2})k_0g + (p^{e-1})k_1g + p_eg(k_2 + \ldots\ldots (k_{e-1})(p^{e-3}))$

$= k_0 (p^{e-2})g + k_1((p^{e-1})g)$

Note that k_0 is already know, hence compute

$h1 = (p^{e-2})h - k_0 (p^{e-2})g$

$= (p^{e-2})(h - k_0g).$

And since $(p^{e-1})g$ has order p, solve the DLP $k1((p^{e-1})g) = h1$ to get the second coefficient of (3).

Then continue in this fashion. Assuming $k_0, \ldots\ldots ki$, are already computed and solve $(k_{i+1})((p^{e-1})g) = (p^{e-i-1})(h - (k_0 + k1p + \ldots\ldots + k_ip^i)g)$ with the assumed algorithm. In total, apply the algorithm e times to obtain all coefficients of (3). Each algorithm takes O(Sp) steps, therefore this methods takes O(eSp) steps.

If we use the Pollard ρ-algorithm, it takes $O(\sqrt{pe})$ steps to solves the DLP if g has order p^e. The method described above can reduce the number of steps to $O(e^{\sqrt{p}})$.This is a significant improvement when $e \geq 2$.

Now suppose kg = h in an abelian group G, with $n = p_1^{e1}.p_2^{e2}\ldots.p_t^{et} = ord_G(g)$

Compute k as follows:

1. For each $1 \leq i \leq t$, Let

And

And solve $k_i g_i = h_i$ using the method described above.

2. Solve the system $k = k1 \pmod{p_1^{e1}}, \ldots k = k_t \pmod{p_t^{et}}$ using the Chinese remainder theorem.

This method is known as the Pohlig-Hellman algorithm. If assume that solving the DLP for some base g with prime order p takes O(Sp) steps, we get the following theorem:

Theorem 2.5.2. If h Є <g>, the Pohlig-Hellman algorithm will solve the DLP in

$O(\sum_{i=0}^{t} eiSpi + \log 2(n))$ steps.

Proof. Suppose x is a solution for the system of congruence's in step 2. Then for every i, we can write $x = x_i + q_i p_i^{ei}$ for some q_i. Compute

$$\frac{n}{piei}(xg) = \frac{n}{piei}((x_i + q_i p_{iei})\, g)$$

$$= \frac{nxi}{piei} g + qing$$

$$= x_i g_i$$

$$= hi = \frac{n}{piei} h_i$$

Hence (n=i)x _ (n=peii)k (mod n), since discrete logarithms are only defined modulo the order of g. Now the numbers $(n/p_1^{e1}), \ldots, (n/p_t^{et})$ are pair wise relatively prime. Hence we can find $C_1, \ldots, C_t$ such that

From $(n/p_{iei})x \equiv (n/p_{iei})k \pmod{n}$ for all I

And from this conclude that $x \equiv k \pmod{n}$. So the algorithm indeed produces a discrete logarithm of h to base g.

In step 2 the system of congruences can be solved in O(log2(n)) steps and solving each DLP of step 1 can be solved in $O(e_i S_{pi})$ steps. So indeed, the Pohlig-Hellman algorithm takes $O(\sum_{i=1}^{t} e_i S_{pi} + log^2(n))$ steps.

Index calculus in F_p^*

For general groups, the only algorithm known to solve the DLP takes exponential time. For some specific groups however, there exist faster algorithms. We will now describe index calculus on F_p^*, which solves the DLP in sub-exponential time.

For details we refer to Section 4.6 of [4].

First we need some definitions:

Definition Let n,B Є N. Then n is called B-smooth if every prime factor of n is smaller than or equal to B. If x Є (Z=nZ)*, then x is called B-smooth if its smallest positive representative in N is B-smooth.

Definition Let : N ! N be the function that assigns to each n Є N the number of primes smaller than or equal to n.

Let g Є F_p^* be an element of order p-1. Our first goal is to determine $\log_g(l)$ for small primes l. Let us assume that g_i is the smallest positive representative of g^i. If g^i is B-smooth for some number B, that can be written as:

$$g^i \equiv g_i \equiv \prod_{l \le B} l^{el(i)} \pmod{p}$$

and therefore,

$$I \equiv \prod_{l \le B} e_l(i).\log l \pmod{p-1}$$

Note that this gives a linear equation in `$\log_g(l)$' with l prime. So if the number of B-smooth g^i's exceeds (B), we get a system of linear equations with a unique solution. We expect to find (B) numbers that are B-smooth in sub exponential time.

Once we have determined the `$\log_g(l)$', finding k such that gk = h in F_p^* is easy. We first search for a j, with $0 < j < p - 1$, such that hg^{-j} is B-smooth. We only need one j, so we expect to find it quite fast. The B-smoothness implies $hg^{-j} \equiv \prod_{l \le B} l^{el} \pmod{p}$.

Thus

$$\log_g(h) \equiv j + \prod_{l \le B} e_l.\log_g(l) \pmod{p-1}$$

And the discrete logarithm is founded due to need.

The method described above also works in other groups, as long as the concept of B-smoothness exists. According to [8], there is also no index calculus possible in F_{pk}^* if $p > 2$ and $k > 1$.

3.6.4 Consequences for cryptography

In this Section four algorithms are describe that solve the discrete logarithm problem in general groups. There are no known algorithms that solve the DLP for general groups in polynomial time. But the exponential time algorithms do place some restrictions on the groups, elements and exponents For example, as already mentioned that the naive algorithm forces to use elements with very large order. The order should be > 280 according to [6]. Also, the exponent needs to be huge. But the number of steps needs to solve DLP is reduces by the baby-step giant-step algorithm and the Pollard ρ-algorithm dramatically. So in order to keep cryptosystems secure, the order

of the base needed to be dramatically larger, > 2160 to be precise. That's not all. The Poling-Hellman algorithm also incenses our choices. When the order of the base element is a product of small primes, this algorithm makes the DLP quite simple to solve. Therefore this process should have at least one enormous factor, larger than 2160. It is also a bad idea to use F_p^* for cryptographic purposes, especially when p is small. Since then index calculus will give a discrete logarithm relatively fast. Of course, the speed of the algorithms described is relative. DLP requires long time to solve it. But there are some algorithms that finish exceptionally fast. For example, the running time of the Pollard ρ-algorithm depends on chance. Maybe some lucky shot, a discrete logarithm is solved in reasonable time by it. Furthermore, computers get faster every day. The time an algorithm needs to finish depends on the speed of computer calculations. Therefore, algorithms become faster. So the lower limit for order and exponent needs to be increased regularly.

3.7 Application of public key cryptography:

There are two main use of PKC.

3.7.1 Confidentiality: This is an application of a public key encryption system in which a message being encrypted before sending and refer to decrypt after receiving the message using the recipient's public key and recipient's paired private key respectively [35]. This assumes, there is no flaw that is discovered in the basic algorithm used.

3.7.2 Digital Signature: Digital signature is another application of public-key cryptography. This scheme helps to authenticate the sender and also used for non-repudiation. To provide authentication, sender generate a digital signature of the message and sends it along with the message to the intended receiver [35]. To provide

confidentiality the whole message with digital signature is encrypted using the recipient's public key.

Many other cryptographic protocols and applications are constructed using these characteristics, such as digital cash, password-authenticated key agreement, multi-party key agreement, time-stamping services, non-repudiation protocols, etc.

CHAPTER 4
ELLIPTIC CURVE

4.1 Introduction of Elliptic Curve:

An **elliptic curve** (**EC**) is a flat, projective algebraic curve of one of its kind, on which there is a specified point O which serves the identity element [36]. Any elliptic curve can be written as a plane algebraic curve defined by an equation of the form:

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

which is non-singular; that is, generated graph using that equation does not has any cusps or self-intersections. The point *O* is represented as "point at infinity" in the plane.

If equation of curve is represented as $y^2 = P(x)$ and if *P* is a polynomial of degree three in *x* with no roots being repeated, then generate a non singular curve, which is an elliptic curve. If *P* is represented as a polynomial of degree four then generated curve is square free and again describes a plane curve of one of its kind; however, it does not have an option of identity element. It is *not* an ellipse. An elliptic curve is a set of points on a plane which satisfy an equation of the form $y^2 = x^3 + a\,x + b$. For an instance an elliptic curve $y^2 = x^3 - 3x + 5$: that is described by the figure 4.1

The elliptic curve is the set of points which satisfy an equation like that. For the curve described below that points (1, 1) and (1, -1) lies on the curve and a mental calculation confirms that they fit into the equation. But the points on an elliptic curve form a structure which is sufficient to form a group. Being a group four things requires to be satisfied by points: closure to group, follow associative rule, contains inverse element and identity element also. If resultant of operation is exists in the group then it is closure to given group. This is true for all elements: (a + b) + c = a + (b + c). Group contains a zero element so that

'a' + 0 = 'a' that is identity element for addition. Finally that, for every element, there's a negative of that element (written -a), so that a + -a = 0.

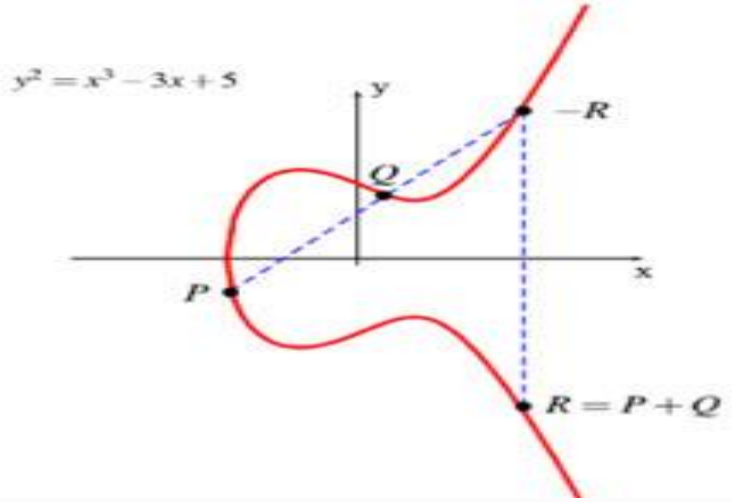

Fig-4.1 Elliptic Curve

4.2 The Group Law:

By adding up a point at infinite location, the projective description of this curve is accomplished. If two points P and Q are on the curve, then third point can be distinctively depicted that is an intersection of the curve by the line passing through P and Q. If a line passing through a point is tangent to the curve then point is counted two times: and if line is parallel to y-axis then the third point is defined as the point "at infinity". Hence, for any pair of points on an elliptic curve, one of the above condition holds good [36].

Group operations can be introduced on curve for '+' operation with the following properties: 0 is supposed to be the point at infinity; and if the points P, Q and R are intersected by a straight line, then require that in the group P + Q + R = 0. It can be used to

check that whether the curve can be turned into an abelian group and into an abelian variety or not. In figure 4.2 operations performed on group are represented.

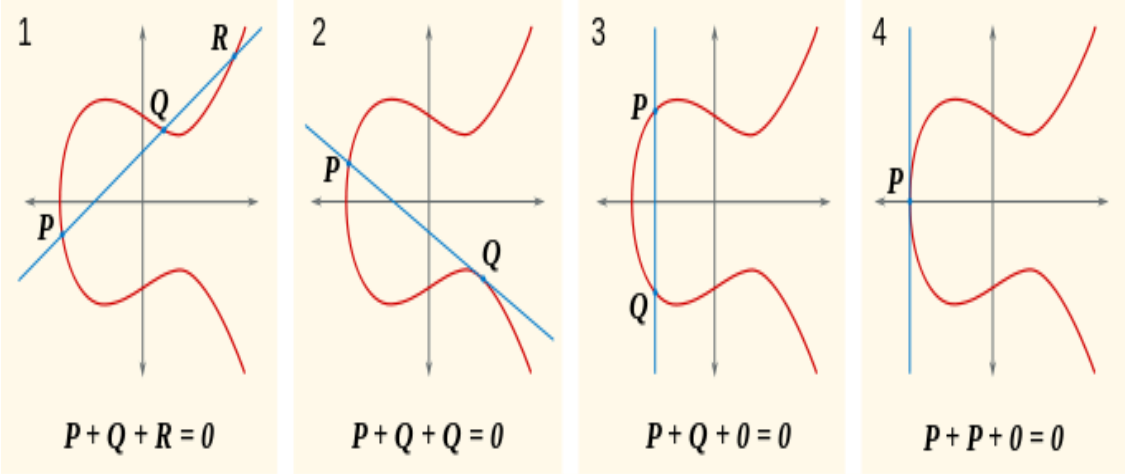

Fig 4.2 Group operations

The above group can be described algebraically as well as geometrically. Given the curve $y^2 = x^3 - px - q$ over the field K and points $P = (X_P, Y_P)$ and $Q = (X_Q, Y_Q)$ on the curve, assume first that $X_P \neq X_Q$. Let s be the slope of the line containing P and Q; i.e.,

$$s = \frac{y_P - y_Q}{x_P - x_Q}.$$

Since K is a field, s is well-defined. Then we can define $R = P + Q = (X_R, -Y_R)$ by

$$x_R = s^2 - x_P - x_Q$$
$$y_R = y_P + s(x_R - x_P).$$

If $X_P = X_Q$, then there are two options: if $Y_P = -Y_Q$, including the case where $Y_P = Y_Q = 0$, then the sum is taken to be 0 and hence by reflecting the curve along the x-axis, the inverse of each point on the curve is evaluated. If $Y_P = Y_Q \neq 0$ (second pane), then $R = P + P = 2P = (X_R, -Y_R)$

is given by

$$s = \frac{3{x_P}^2 - p}{2y_P}$$
$$x_R = s^2 - 2x_P$$
$$y_R = y_P + s(x_R - x_P).$$

Associatively: The sum of the three values on any of the six lines is zero. The location of all nine points along with the location of a, b, c and zero is found with the help of elliptic curve. The central point of the nine lies on the line through a and b + c, and also on the line through a + b and c. Associativity of the addition law is equivalent to the fact that the curve passes through the central point in the grid. From this fact, the equality of −(a + (b + c)) and −((a + b) + c) follows [36].

The elliptic curve and the point zero are kept constant in this simulation while a, b and c move independent of each other.

4.3 Elliptic Curve Over General Field:

EC can be defined over any field K; the formal definition of an elliptic curve is a non-singular projective algebraic curve over K with genus 1 with a given point defined over K. If the characteristic of K is neither 2 nor 3, then the equation of elliptic curve over K can be written in the form

$$y^2 = x^3 - px - q$$

Where p and q are elements of K such that the right hand side polynomial $x^3 - px - q$ does not have any double roots. More terms need to be kept if the characteristic is 2 or 3, whereas if characteristic is 3, the most general equation is of the form

$$y^2 = 4x^3 + b_2x^2 + 2b_4x + b_6$$

for arbitrary constants b_2, b_4, b_6 such that the polynomial on the right-hand side has distinct roots. Even this much is not possible when the characteristic is 2, and the most general equation is

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

provided, that the variety it defines is non-singular. If characteristic was not a hindrance, each equation would lessen down to the previous ones by a suitable change of variables [37]. One typically takes the curve to be the set of all points (x, y) which satisfy the above equation and such that both x and y are elements of the algebraic closure of characteristic K. K-rational points are the points of the curve whose both coordinates belong to K.

4.4 Elliptic Curve Group Over Real Number:

An EC over real numbers may be defined as the set of points (x, y) which satisfy an

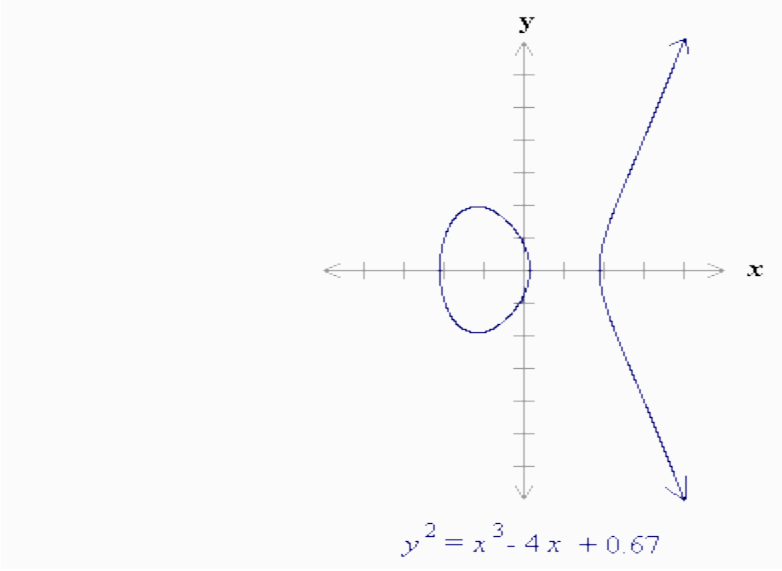

Fig-4.3 elliptic curve with equation y2 = x3 - 4x + 0.67

elliptic curve equation of the form:

y2 = x3 + a x + b, Where x, y, a and b are real numbers.

Each choice of the numbers 'a' and 'b' yields a different elliptic curve. For example, a = -4 and b = 0.67 gives the elliptic curve with equation $y^2 = x^3 - 4x + 0.67$; the graph of this curve is shown below: If $x^3 + a\ x + b$ contains no repeated factors, or equivalently if $4a^3 + 27b^2$ is not 0, then a group can be formed using the elliptic curve $y^2 = x^3 + a\ x + b$. The group formed by the elliptic curve over real numbers comprises of the points on the corresponding elliptic curve along with a special point O known as the point at infinity.

4.5 Elliptic Curve Group Over Finite Field Fp:

Calculations over the real numbers are time-consuming and erroneous due to round-off error. Cryptographic applications require quick and accurate arithmetic calculations; thus elliptic curve groups over the finite fields of F_p and F_2^m are used in practice [37].

Recall that the field F_p uses the numbers from 0 to p - 1, and computations stop by taking the remainder after dividing by p. For example, in F_{23} the field comprises of integers ranging from 0 to 22, and any operation applied within this field will result in an integer that would also lie between 0 and 22.

An elliptic curve with the underlying field of F_p can form by choosing the variables 'a' and 'b' within the field of F_p. The elliptic curve includes all points (x, y) which satisfy the elliptic curve equation modulo p (where x and y are numbers in F_P) [36]. For example:

$y^2 \bmod p = x^3 + ax + b \bmod p$

has an underlying field of F_p if a and b are in F_p.

If $x^3 + ax + b$ contain no repeating factors, then a group can be formed with the help of elliptic curve.

Definition Let K = Fq be the finite field with q elements and E an elliptic curve defined over K. In general it is difficult to calculate the precise number of rational points of an elliptic curve E over K [36], the following estimate can be done with the help of Hasse's theorem on elliptic curves which also gives us the point at infinity:

$$|\mathrm{card}E(K) - (q+1)| \leq 2\sqrt{q}.$$

The set of points E(Fq) is known to be the finite abelian group, which is always cyclic or can also be the product of two cyclic groups.[18] The curve can be defined with the help of equation

$$y^2 = x^3 - x$$

Over F_{71} has 72 points, 71 affine points including (0, 0) and one point at infinity over this field, whose group structure is given by Z/2Z × Z/36Z. The number of points on a specific curve can be computed with Schoof's algorithm that is defined n section 4.6.

Studying the curve over the field extensions of Fq is facilitated by the introduction of the local zeta function of E over Fq, defined by a generating series defined by a generating series

$$Z(E(K),T) \equiv \exp\left(\sum_{n=1}^{\infty} \mathrm{card}\left[E(K_n)\right]\frac{T^n}{n}\right)$$

Where the field Kn is the (unique) extension of K = Fq of degree n (that is, Fqn). The zeta function is a rational function. There exist an integer 'a' such that the equation

$$Z(E(K),T) = \frac{1 - aT + qT^2}{(1-qT)(1-T)}.$$

Moreover,

$$Z\left(E(K), \frac{1}{qT}\right) = Z(E(K),T)$$
$$\left(1 - aT + qT^2\right) = (1-\alpha T)(1-\beta T)$$

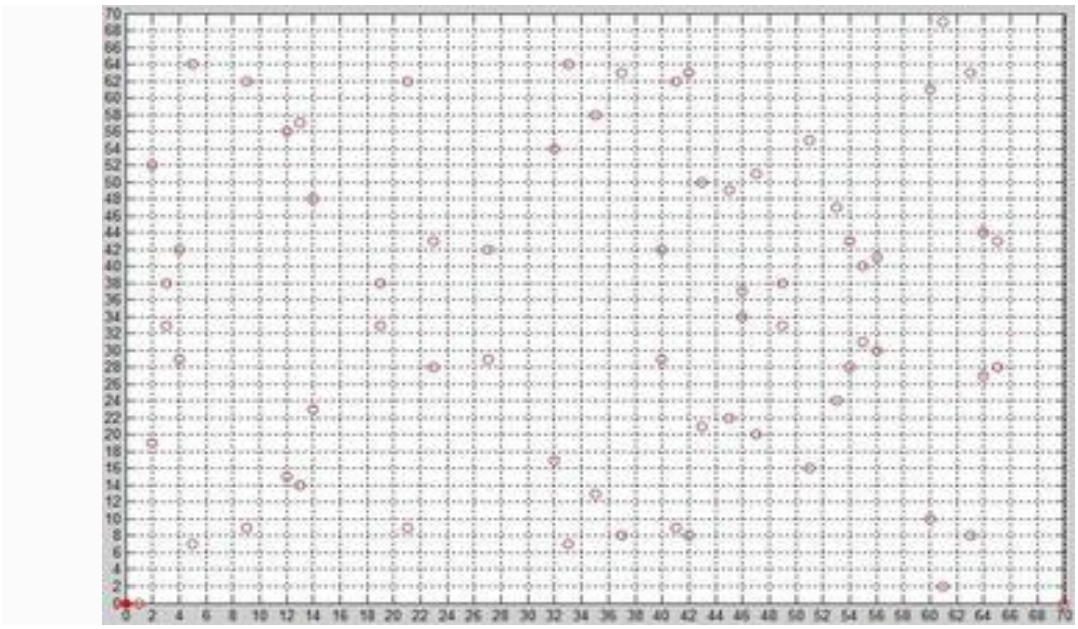

Fig- 4.4 Set of affine point over finite field F_{71}

With complex numbers α, β of absolute value $\sqrt{q}$ and hence, this result is a particular case of the Weil conjectures. The zeta functions of E over the field F_2 is given by the equation:

$$\frac{1+2T^2}{(1-T)(1-2T)}$$

This follows from:

$$|E(\mathbf{F}_{2^r})| = \begin{cases} 2^r + 1 & r \text{ odd} \\ 2^r + 1 - 2(-2)^{\frac{r}{2}} & r \text{ even} \end{cases}$$

Elliptic curves over finite fields are notably applied in cryptography and for the factorization of large integers. These methods frequently make use of the group structure on the points of E [36]. Algorithms that are applicable to general groups can also be applied to the group of points on an elliptic curve and one of its kinds is the discrete logarithm algorithm. The significance is that selecting an elliptic curve allows for more flexibility than choosing q. Also, the group structure of elliptic curves is generally more complicated.

4.6 Schoof's Algorithm

Schoof's algorithm is an efficient algorithm to count points on elliptic curves over finite fields. This algorithm can be applied in elliptic curve cryptography where it is essential to know the number of points to judge the difficulty of solving the discrete logarithm problem in the group of points on an elliptic curve [38].

Previous to Schoof's algorithm, various approaches to counting points on elliptic curves such as the naive and baby-step giant-step algorithms were, for the most part, tedious and had an exponential running time.

Definition. Let E be an elliptic curve defined over the finite field Fq, where $q=p^n$ for 'p' a prime and n an integer ≥ 1. of characteristic ≠ 2, 3 an elliptic curve can be given by a (short) Weierstrass equation

$$y^2 = x^3 + Ax + B$$

With A,B Є Fq [38]. The set of points defined over Fq consists of the solutions (a, b) Є Fq satisfying the curve equation and a point at infinity O. Using the group law on elliptic curves restricted to this set one can see that this set E(Fq) forms an abelian group, with O as the zero element. We compute the cardinality of E(Fq) to count the number of points on elliptic curve. Schoof's approach to computing the cardinality # E(Fq) makes use of Hasse's theorem that is describe in section 4.7, on elliptic curves along with the Chinese remainder theorem and division polynomials [39].

4.7 Hasse's theorm :

Hasse's theorem states that over the finite field Fq, if E/ Fq is an elliptic curve, then

E(Fq) satisfies the equation [38]

$$| q + 1 - \sharp E(\mathbb{F}_q) | \leq 2\sqrt{q}.$$

Defining t to be q+1- #E(Fq), and making use of the obtained result, the calculation of the cardinality of t modulo N where N>4$\sqrt{q}$, is sufficient for determining t, and thus #E(Fq).

CHAPTER 5

Elliptic Curve Cryptography

5.1 Introduction of Elliptic Curve Cryptography (ECC):

Elliptical curve cryptography (ECC) is one of the public key encryption technique that refers elliptic curve theory to create quicker, minor, and more efficient keys. Properties of the elliptic curve to generate keys rather than traditional method of generation of keys which prefer to generate keys using products of large prime numbers that mostly used in combination of encryption methods, such as RSA and Diffie-Hellman. ECC required small key in size to provide a high level of security while others required a large key size for same level of security for example ECC needs 64 bits while other requires 1024 bit for same security. ECC used in mobiles due to its lower computational power and battery resources usage [6].

Victor Miller (IBM) and Neil Koblitz (University of Washington) was proposed an alternative mechanism to implement public-key cryptography in 1985 that is Elliptic Curve Cryptography (ECC). Public-key algorithms help to distribute keys among large number of users in complex information system. ECC is based on discrete logarithm that is much more difficult to challenge at equivalent key lengths [7].

Asymmetric cryptography is a marvellous technology. It uses in many applications and varied such as used in distributed network environments, required during communications to provide secure communication, for reducing key distribution issues with a public key infrastructure (PKI). For designing or employing any network protocol or application requiring secure communications, for a practical solution asymmetric cryptography must be used. Every time when you buy something on the Internet, asymmetric cryptography is used to provide a secure transaction. If asymmetric cryptography is needed then you should concern about methodology that required less resources and ECC is the best choice, because:

• ECC offers significantly better security with a given key size

• The smaller key size is needed for providing given security which can be applied on smaller chips and more compact software and these are able to run faster cryptographic operations that produce less heat and required less power.

•Some efficient and compact hardware implementations are available for ECC operations that offer potential reduction in implementation of footprint even outside of those due to the smaller key length alone.

In short: asymmetric cryptography is in demand. But if due to security reasons ECC has own place for providing better security as compare to other methodologies. This thesis describes elliptic curve cryptography in greater depth. ECC provides considerably large security with a given key size [9].

5.2 Cryptography Premise:

The entire security of ECC depends on the ability to compute a point multiplication and the inability to compute the multiplicand given the original and product points.

5.2.1 Point Multiplication:

Point multiplication operation is not a simple arithmetic operation. This operation is used to performed a transformation of an affine point into coordinate form in which a scalar k is multiplied with another point P that lies on elliptic curve to produce another point Q on the same elliptic curve i.e. k*P=Q

Point multiplication is performed by applying two basic elliptic curve operations [7]

• Point addition, adding two points P_1 and P_2 to produce another point P_3 i.e., $P_3 = P_2 + P_1$.

• Point doubling, adding a single point P_1 to itself to produce another point P_3 i.e. $P_3 = 2P_2$.

Point addition and doubling are discussed in sections 5.2.2 and 5.2.3 respectively

A simple example is explained here to give a brief introduction of point multiplication operation.

Let P be a point on an elliptic curve. Take a scalar point k that is multiplied with the point P to obtain another point Q on the curve. i.e. to find Q = kP [42].

If k = 23 then kP = 23.P = 2(2(2(2P) + P) + P) + P [42].

Thus point multiplication is performed by applying point addition and point doubling repeatedly to compute and produce the result. The above method is called 'double and add' method for point multiplication.

5.2.2 Point Addition

Point addition is an operation that performs addition of two points J and K that lies on an elliptic curve to compute another point L on the same elliptic curve.

Geometrical Explaination:

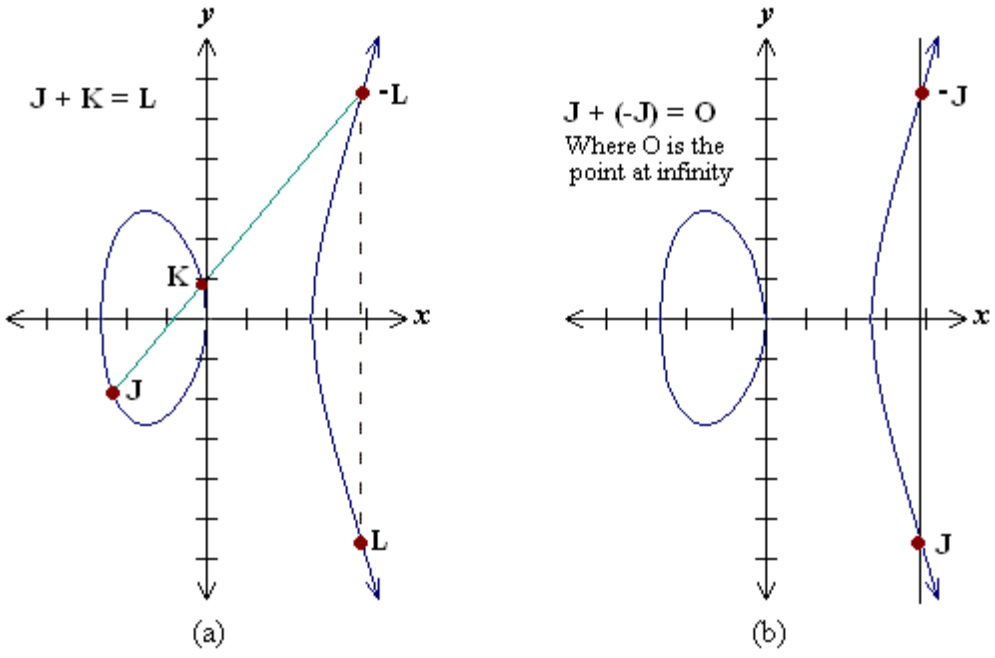

Fig- 5.1 Point Addition Operation

Consider two points J and K on an elliptic curve as shown in figure (a). If K ≠ -J then a line drawn through the points J and K will intersect the elliptic curve at exactly one more point –L. The reflection of the point –L with respects to x-axis produces [7] another point L, which contains the result of addition operation of points J and K.

Thus on an elliptic curve L = J + K.

If K = -J the line through this point intersect at a point at infinity O. Hence J + (-J) = O. This is shown in figure (b). O is the additive identity of the elliptic curve group. A negative of a point is the reflection of that point with respect to x-axis [7].

Analytical explanation

Consider two distinct points J and K such that J = (X_1, Y_1) and K = (X_2, Y_2)

Let L = J + K where L = (X_3, Y_3), then

$X_3 = S^2 – X_1 – X_2$

$Y_3 = -Y_1 + S (X_1 – X_3)$

$S = (Y_1 – Y_2)/(X_1 – X_2)$, S is the slope of the line through J and K.

If K = -J i.e. K = $(X_1, -Y_1)$ then J + K = O. where O is the point at infinity.

If K = J then J + K = 2J then point doubling equations are used.

Also J + K = K + J

5.2.3 Point Doubling Operation:

Point doubling is an operation that performs addition of a point J to itself that les on the elliptic curve to produce another point L on the same elliptic curve.

Geometrical explanation

To double a point J to get L, i.e. to find L = 2J, consider a point J on an elliptic curve as shown in figure (a). If y coordinate of the point J is not zero then the tangent line at J will intersect the elliptic curve at exactly one more point –L. The reflection of the point –L with

respect to x-axis gives another point L, which contains the result of doubling the point J. Thus L = 2J.

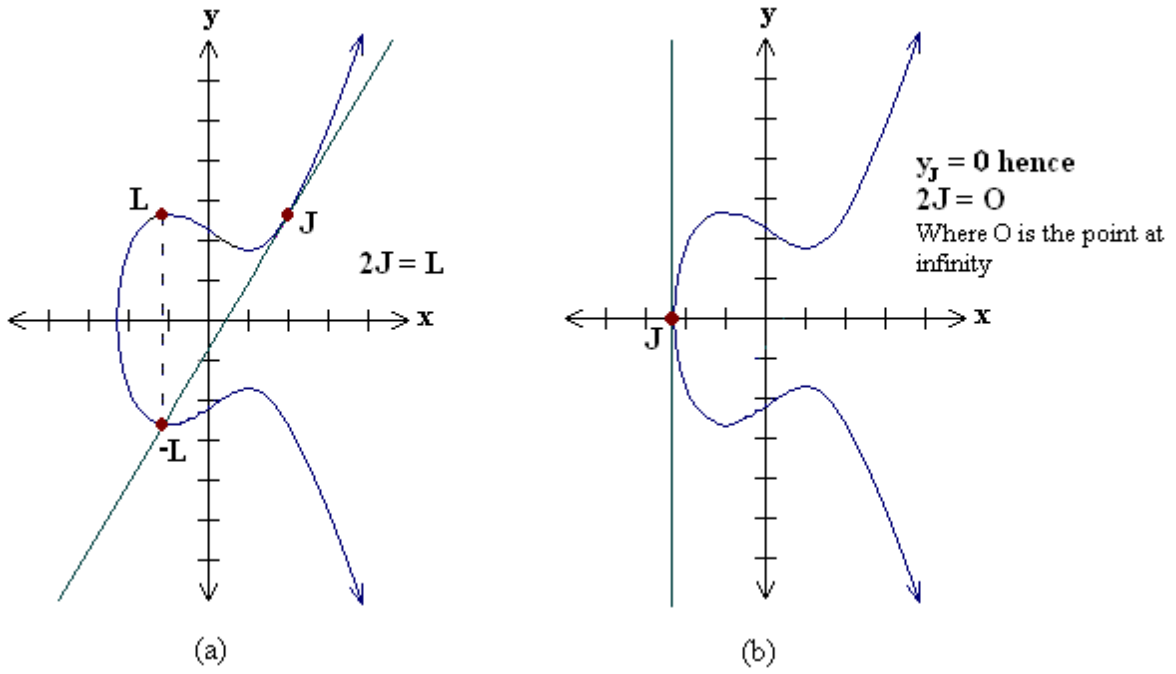

Fig- 5.2 Point Doubling Operation

If y coordinate of the point J is zero then the tangent at this point intersects at a point at infinity O. Hence 2J= O when $Y_1 = 0$. This is shown in figure (b) [7].

Analytical explanation

Consider a point J such that J = (X_1, Y_1), where $Y_1 \neq 0$

Let L = 2J where L = (X_3, Y_3), Then

$X_3 = S^2 - 2X_1$

$Y_3 = -Y_1 + S(X_1 - X_3)$

$S = (3X_1^{\ 2} + a) / (2Y_1)$, S is the tangent at point J and a is one of the parameters chosen with the elliptic curve

If $Y_1 = 0$ then 2J = O, where O is the point at infinity.

5.3 Finite Fields

Finite field is defined as a subset of real numbers. There is round off error occurred for real numbers when operations are performed on it. Due to this problem performance of system is slow and inaccurate. Cryptographic operations required to be performed more rapidly and produce accurate result. Two finite fields are acquiring on which curve cryptography is defined to perform operations more accurately and effectively on ECC.

• Prime field Fp and

• Binary field F_2^m

Choose a field with a large number of primes that is suitable for cryptographic operations. Section 5.4 and 5.5 explains the EC operations on prime fields and binary fields respectively. In these sections affine coordinate system is referred to perform operations in which each point is represented by the vector (x,y).

5.4 EC on Prime field Fp

General equation of curve on a prime field Fp is

$$\mathbf{y^2 \bmod p = x^3 + a\,x + b \bmod p,}$$

where $\mathbf{4a^3 + 27b^2 \bmod p \neq 0}$. Here finite field is also a subset of elements that are integers ranging between 0 and p – 1. All the operations such as subtraction, addition, multiplication and divisions engage with integers within 0 and p – 1. Choose a prime number p in such a way that there are large numbers of points generated on curve that are capable to provide a secure cryptosystem. [4].

The curve generated by this elliptic curve equation is not a smooth curve. Therefore point addition and doubling operation performed on real numbers will not work here. Though, the algebraic rules for point addition and point doubling can be adapted for elliptic curves over Fp.

5.4.1 Point Addition

Consider two distinct points P_1 and P_2 such that $P_1 = (X_1, Y_1)$ and $P_2 = (X_2, Y_2)$

Let $P_3 = P_1 + P_2$ where $P_3 = (X_3, Y_3)$, then

$X_3 = S^2 – X_1 – X_2 \bmod p$

$Y_3 = -Y_1 + S (X_1 – X_3) \bmod p$

$S = (Y_1 – Y_2)/(X_1 – X_2) \bmod p$, S is the slope of the line through P_1 and P_2.

If $P_2 = -P_1$ i.e. $P_2 = (X_1, -Y_1 \bmod p)$ then $P_1 + P_2 = O$. where O is the point at infinity.

If $P_2 = P_1$ then $P_1 + P_2 = 2P_1$ then point doubling equations are used.

Also $P_1 + P_2 = P_2 + P_1$

5.4.2. Point Subtraction

Consider two distinct points P_1 and P_2 such that $P_1 = (X_1, Y_1)$ and $P_2 = (X_2, Y_2)$

Then $P_1 – P_2 = P_1 + (-P_2)$ where $–P_2 = (X_2, -Y_2 \bmod p)$

Point subtraction is used in certain implementation of point multiplication such as NAF [1].

5.4.3. Point Doubling

Consider a point P_1 such that $P_1 = (X_1, Y_1)$, where $Y_1 \neq 0$

Let $P_3 = 2P_1$ where $P_3 = (X_3, Y_3)$, Then

$X_3 = S^2 – 2X_1 \bmod p$

$Y_3 = -Y_1 + S(X_1 – X_3) \bmod p$

$S = (3X_1^2 + a) / (2Y_1) \bmod p$, S is the tangent at point P_1 and a is one of the parameters chosen with the elliptic curve

If $Y_1 = 0$ then $2P_1 = O$, where O is the point at infinity.

5.5 EC on Binary field F_2^m

General equation of elliptic curve for binary field is described below:

$$\mathbf{y^2 + xy = x^3 + ax^2 + b}$$

where $\mathbf{b \neq 0}$.

Finite field consists of elements that are integers and length of these integers is at most m bits while these numbers can be measured in degree of m-1 as a binary polynomial and coefficients are represented by either 1 or 0. In binary polynomial all the operations such as addition, subtraction, multiplication and division are performed in degree m-1 or lesser. Value of m is chosen in such a way that make the cryptography secure. The curve generated by this equation is not a smooth curve; so point addition and doubling cannot be applied over real numbers. Though, these rules are adapted for elliptic curve over F_2^m.

5.5.1 Point Addition

Consider two distinct points P_1 and P_2 such that $P_1 = (X_1, Y_1)$ and $P_2 = (X_2, Y_2)$

Let $P_3 = P_1 + P_2$where $P_3 = (X_3, Y_3)$, then

$X_3 = S^2 + S + X_1 + X_2 + a$

$Y_3 = S (X_1 + X_3) + X_3 + Y_1$

$S = (Y_1 + Y_2)/(X_1 + X_2)$, S is the slope of the line through P_1 and P_2.

If $P_2 = -P_1$ i.e. $P_2 = (X_1, X_1 + Y_1)$ then $P_1 + P_2 = O$. where O is the point at infinity.

If $P_2 = P_1$ then $P_1 + P_2 = 2P_1$ then point doubling equations are used.

Also $P_1 + P_2 = P_2 + P_1$

5.5.2. Point Subtraction

Consider two distinct points P_1 and P_2 such that $P_1 = (X_1, Y_1)$ and $P_2 = (X_2, Y_2)$

Then $P_1 – P_2 = P_1 + (-P_2)$ where $–P_2 = (X_2, X_2 + Y_2)$

Point subtraction is performed in some implementation of point multiplication such as NAF [1].

5.5.3. Point Doubling

Consider a point P_1 such that $P_1 = (X_1, Y_1)$, where $X_1 \neq 0$

Let $P_3 = 2P_1$ where $P_3 = (X_3, Y_3)$, Then

$X_3 = S^2 + S + a$

$Y^3 = X_1^{\,2} + (S + 1)*X_3$

$S = X_1 + Y_1/ X_1$, S is the tangent at point P_1 and a is one of the parameters chosen with the elliptic curve

If $X_1 = 0$ then $2P_1 = O$, where O is the point at infinity.

5.6 Elliptic Curve Domain parameters

There are many parameters apart from 'a' and 'b', that must be approved by both sender and receiver used in secured and trusted communication using ECC [19]. Both binary field and binary have different domain parameters that are described below.

5.6.1 Domain parameters for EC over field Fp

p, **a**, **b**, **G**, **n** and **h** are domain parameters for Elliptic curve over Fp.

Where p = prime number defined for finite field Fp.

'a' and 'b' = parameters that is used to describe the curve $y^2 \bmod p = x^3 + ax + b \bmod p$.

G = generator point (X_G, Y_G), that is a point on curve chosen for cryptographic operations.

n = order of the curve [19].

The scalar for point multiplication is chosen as a number between 0 and n – 1.

h = cofactor where h = #E(Fp)/n. #E(Fp) is the number of points on an elliptic curve.

5.6.2. Domain parameters for EC over field F_2^m

For elliptic curve over F_2^m **a**, **b**, **G**, **m**, **f(x)**, **n** and **h** are domain parameters.

Where m is an integer defined over finite field F_2^m. The elements of the finite field F_2^m are integers of length at most m bits. f(x) is the irreducible in polynomial of degree m used for elliptic curve operations [42]. 'a' and 'b' are the parameters defining the curve $y^2 + xy = x^3 + ax^2 + b$. G is the generator point (X_G, Y_G), a point on the elliptic curve chosen for cryptographic operations [42]. The order of the elliptic curve is n [19]. The scalar value is chosen as a number between 0 and n – 1 for point multiplication. h is the cofactor where h = #E(F2m)/n. #E(F2m) is the number of points on an elliptic curve.

5.7 Discrete Logarithm Problem (DLP):

The security of ECC depends on the difficulty of Elliptic Curve Discrete Logarithm Problem [42]. Let there is two points P and Q that exists on an elliptic curve such that kP = Q, where k is a scalar [42]. If value of P and Q are given, then it is computationally infeasible to obtain k, if k is sufficiently large. Here k is the discrete logarithm of Q to the base P [42].

As discrete logarithm problem is discussed in section 3.4, public key cryptosystems based on DLP is discussed in section 3.5 and different algorithms for solving DLP in section 3.6. This section is included to show that ECC depends on difficulty of DLP and how it helps to provide a security and a secure communication between two users.

5.8 Advantages-

a) Key size and digital signature that are generated through ECC are very shorter in size compare to other Cryptography scheme.

b) This is based on discrete logarithmic form so easily can be converted into elliptic curve form.

c) No time consumes for permutation and combination and less time taking for encryption.

d) Till date no solution found for breaking the Discrete Logarithmic approach so brute force attack on ECC takes too many years (uncountable).

e) Very much suitable for handheld devices such as palm top mobile phones PDA because they are low memory devices and ECC can work better on this.

5.9 Disadvantages-

a) ECC uses curves generators fields' etc. This is more complex to calculate so this is not good for processor health.

b) ECC systems are much slower than RSA in large no. of public key generation.

c) For performing calculations on more complex variables so it is also not good for device's resources such as memory, processor etc.

5.10 Applications-

a) Simple Key generation by ECC is a great application in cryptography.

b) Shorter Certificate

c) Shorter Signature can also be generated with the help of ECC.

Generally till date ECC worked only in constrained environment such as less memory shorter devices and limited ROM and limited processing speed so it may be our new future work to make ECC more independent from system and devices and constrained environment.

Now we are moving to the IPV6 because there are some drawbacks in IPV4 such as when transfer data over network then IPSEC protocol and IPV4 can't work simultaneously so it is unable to provide a secure transmission. To make a secure transmission and make it better for doing a better work, elliptic curve cryptography is the solution for this because when data is transferred by using the RSA (Rivest-Shamir-Adleman) algorithm then a long key in size is required for encryption and decryption function and there may be possibility that data becomes corrupted and inconsistent in middle of the way but elliptic curve cryptography reduces size of key as compare to RSA and the transmitting data also become safe so that no one can become corrupted data and it provides less possibility of altering of data through by adversary [17].

CHAPTER 6
PROPOSED WORK

6.1 Elliptic curve cryptography (ECC):

The proposal of using Elliptic curves in cryptography was introduced by Victor Miller and N. Koblitz as an alternative to established public-key systems such as DSA and RSA [18]. In Elliptical curve Discrete Log Problem (ECDLP) makes it difficult to break an ECC as compare to RSA and DSA where the problem of factorization or the discrete log problem can be solved in sub-exponential time [18]. This means that ECC considerably utilizes smaller parameters rather than other systems like RSA and DSA. So ECC being capable in having a key small in size consequently leads to faster computations.

Elliptic curve is already introduced in section 4 and introduced about ECC in chapter 5 so don't need to discuss again about elliptic curve and ECC. This chapter tell about how do ECC work and implement on a system? How all the operations like point multiplication, addition, and doubling is performed on a message and when? How users distribute their keys on network?

Elliptic curve is used by elliptic curve cryptography which restricts to all the variables and coefficients to be elements of a finite field. This procedure is initiated with an affine point in ECC called Pm(x,y). These affine points may be some other point nearest to the Base point (G) or to be the Base point itself. Base point refers to the smallest (x,y) co-ordinates, which satisfy the EC [18].

Take an example let users A and B want to communicate and they know about ECC. They start with elliptic curve and generate points that help to generate their keys. How these points help them? After generating point's user A choose a random number that is a private key and keeps secret. As user A, user B also choose a random number and used as a private key and also keeps secret. Let user A choose 13 and user B choose 15 and perform multiplication

operation on base point, which is a smallest point of curve, with private key let base point is G (1, 1). Multiplication is performed by performing point addition and doubling continually.

Hence (private key).(base point)=13G=(G+2(2(G+2G)))

So 2G is representing doubling operation. Why? Here a base point G is added with self to generate another point. And then resultant is added with base point G and generates another result and this process will be continuing recursively until it does not reach to parent calling function.

6.2 Generate Public and Private Key and key distribution:

User A's (or User B's) public and private keys are associated with a particular set of elliptic key domain parameters {p, Fp, a, b, G, n, h} where

• p: prime power, p=q or p= 2^m , where q is a prime

• Fp: field representation of the method used for representing field elements ∈ Fq

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

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

• G: A base point represented by G= (Xg, Yg) on E (Fq)

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

• h: cofactor, and is equal to the ratio #E(Fq)/n, where #E(Fq) is the curve order

Private Key is a random number which is generated by user and keep secretly and public is generated by multiplying a base point G with private key of user. Let private key of user A is n_A and public key of user A is generated as $P_A=n_AG$ if private key n_A is 15 then

$P_A=15(G)=G+2(G+2(G+2G))$

After generating keys public key is distributed over network which is used for communicating user for the encryption of message so confidentiality become remain until someone does not get private key of user that is so difficult to obtain from public key because that multiplication is not arithmetic operation, it's point multiplication which is performed by performing addition and doubling repeatedly. User B also follows the same procedure as user A follows and generates public and private key. Both share their public keys over secure communication and generate secret key so that they can share their keys easily [18].

6.2.1 Mathematical Analytical:

Let user A and user B want to communicate then

Global Public Element:

$E_p(a,b)$ elliptic curve with parameter a, b, and p in the equation

$$Y^2 \bmod p = (X^3 + aX + b) \bmod p$$

Q Base point on elliptic curve

User A key generation:

Select private key n_A $n_A < n$

Calculate public key P_A $P_A = n_A G$

User B key generation:

Select private key n_B $n_B < n$

Calculate public key P_B $P_B = n_B G$

Generation of secret key by user A:

$$S1 = K = n_A P_B$$

Generation of secret key by user B:

$$S2=K=n_B P_A$$

These both calculation generate same result because

$$n_A P_B = n_A n_B G = n_B (n_A G) = n_B P_A$$

To crack this system, if G & kG is given then an attacker would require to be able to obtain k, which is tough to determine.

For example, scalar multiple k is 5; G ≡ (2,2) then let 5G= D≡(153,108) for a=0, b=-4, q=211. It is difficult to find out the scalar multiple k=5 [18], the values of G and D are given.

6.3 Elliptic Curve Encryption and Decryption:

(1) Let a message 'm' sending from user A to user B. User A chooses a random positive integer 'k', a private key 'n_A' and generates public key $P_A=n_AG$ and produces the cipher text 'Ct' is a pair of points $Ct =\{kG, Pm+kP_B\}$.

Where G is the base point lies on elliptic curve and selected from generated points, $P_B=n_BG$ is the public key of User B with private key 'n_B'.

(2) To decrypt the cipher text, user B multiplies the 1st point in the pair by user B's secret key and subtracts the result from 2nd point

$$Pm + kP_B - n_B(kG) = Pm + k(n_B G) – n_B(kG)=Pm$$

(3) Then apply decrypt logarithm problem on Pm and get the original message m.

6.4 Implementation of encryption procedure:

Before transmitting image on network encryption of image is needed so to perform encryption process first generate points on elliptic curve and then keys are generated that are

so helpful in encryption and decryption of an image. After that perform encryption in which scalar values are transformed into affine values.

6.4.1 Generate points on elliptic curve:

There is a regular need to maintain a database for points that satisfy the elliptic curve equation, for generating points follow the code mentioned below to check all Y co-ordinates for specified X co-ordinates that satisfy equation has been incorporated. Equation of elliptic curve is given below:

$$y^2 \bmod p = (x^3+ax+b) \bmod p$$

Where, p is a prime number.

Algorithm input a, b, p

Step 1 take x=0 or any other positive integer

Step 2 loop until x<p

I. $Y^2=(X^3+aX+b) \bmod p$

II. If Y^2 is perfect square

Print(X, square root (Y))

Else

X=X+1

Step 3 End.

For example p=37 a=-1 b=1

X	Y
1	1

3	24
6	1
8	30
14	3
24	27
25	22
31	1
32	17
33	29
36	21

Table 6.1 elliptic points

6.4.2 Code to find public key:

Input: XG=X co-ordinate of G

YG=Y co-ordinate of G

nA be the private key. (A scalar multiple)

Let PA be the public key. PA= nA×G

We perform recursive addition of point G for nA times to get the point PA.

For example: G= (2, 2), nA=5, (153,108) = 5(2, 2). So, PA= (153,108) is the public key for [18] private key 5.

6.4.3. Multiplication operation:

This is an optimized approach of multiplication operation, which reduces required memory spaces to save state of calling function and their variables in recursion approach, which is based on iterative method in which we just need some iteration to perform operation. In this approach first we generate binary code of scalar value and number of iteration equals to

(number of bits-1) to represent scalar value. If bit value is zero (0) then perform only doubling operation and if bit value is one (1) then perform both doubling and addition operation.

Example

7*(1, 2) then

7*(1,2)=(1,2)+2((1,2)+2(1,2)) here number of doubling operation=2 and number of addition= 2 mean total number of operation= 4 using recursion.

Using iteration-

Binary of 7= 111 number of bits =3 so number of iteration= 2

In first iteration bit value is 1 so perform addition and doubling and then in second iteration bit value is 1 so again perform addition and doubling so total number of operation= 4 but number of iteration is reduced.

Algorithm input k, B

```
Funmul(k,B) // here k is scalar value and B is any coordinate value
{
    Binary=convert scalar value in to binary
    For 1=1 : binary-1
        If  binary(i)==0
            Perform doubling operation
        Else
```

Perform addition and doubling operation

End

end

}

6.4.4 Encryption:

Elliptic curve cryptology is applied on an image that is a transformation of an image into affine points lie on elliptic curve by performing multiplication operation. Let a is a scalar value representing to a pixel value of an image then $P_{ML}=a* P_M$ yield a coordinate value that is a transformation of scalar value in to affine point and it is evaluated by performing multiplication on affine point P_M with pixel value. After that P_{ML} is added with KPU_B where K is a random number and PU_B is a public key of user B. Completion of encryption generates cipher text $CF=\{KG,P_{ML}+KPU_B\}$ here first part KG formed a coordinate form mean (x1,y1) and second part of cipher text $P_{ML}+KPU_B$ is also formed a coordinate form i.e. (x2,y2) and finally cipher text is CF={(x1,y1),(x2,y2)}. This is an encrypted data that is yield by encryption procedure.

Algorithm input c, P_m, N_B, G

EccEncrypt (c, P_m, N_B, G)

Step 1-For all c (i.e. pixel value)

Find $P_{ml}=c *Pm$ // c is pixel value, P_m is random point in elliptic curve

Step 2- Find $PU_B=PR_B*B_P$ //B_P is the base point Of Elliptic curve,

Step 3- End;

Encrypted data= (kB_P, Pml+k*PU_B)

6.5. Implementation of Decryption:

On receiver side receiver receives cipher text i.e. non readable form of message it is a combination of two pair of coordinate kB_P and Pml+k*PU_B. kB_P is first point of cipher text and Pml + k*PU_B is second point of cipher text then perform multiplication on first part with private key of user.

kB_P *PR_B =first point * private key of user B and obtain result that is subtracted from second point i.e.

Pml + k*PU_B – k*B_P*PR_B=Pml

After getting Pml i.e. Pml=c*Pm from cipher text apply discrete logarithm problem and get original message i.e. c.

6.5.1 Discrete logarithm problem:

Algorithm input r0,r1 that is x-coordinate and y-coordinate [4]

```
decryption(r0,r1)
   g=1;
   while(rem(r0,2)==0 && rem(r1,2)==0)
   r0=r0/2;
   r1=r1/2;
   g=2*g;
   end
```

```
u=r0;
v=r1;
a=1;
b=0;
c=0;
d=1;
count=1;
while(count)
   while(rem(u,2)==0)
      u=u/2;
      if(rem(a,2)==0 && rem(b,2)==0)
      a=a/2;
      b=b/2;
      else
      a=(a+r1)/2;
      b=(b-r0)/2;
      end
end
      while(rem(v,2)==0)
```

```
    v=v/2;
    if(rem(c,2)==0 && rem(d,2)==0)
    c=c/2;
    d=d/2;
    else
    c=(c+r1)/2;
    d=(d-r0)/2;
    end
end
if(u>= v)
u=u-v;
a=a-c;
b=b-d;
else
v=v-u;
c=c-a;
d=d-b;
end
if(u==0)
```

```
                gcd=v*g;
                   if(gcd==1)
                   fprintf('in verse is exist and it is %d mod %d i.e. )',d,r0);
            count=0;
              r=mod(r0,d);
             fprintf('%d',r);
            return ;
            end
          else
            fprintf('\n \n value value value of d and r0 is %d   %d ',d,r0);
            count=1;
             r=0;
         end
      end
```

This algorithm helps us to perform invertible of encryption operation.

CHAPTER 7
RESULT AND ANALYSIS

Elliptic curve cryptology is performed by taking elliptic curve. A general Elliptic Curve is taken that is represented by the following equation:

E: $Y^2=(X^2+aX+b) \bmod p$

Where X, Y are elements of GF(p) and a, b are integers modulo p, satisfying :

$4a^3+27b^2 \neq 0 \pmod p$

Generate elliptic curve values for which a=-1, b=1 and p=37

Generated values are given that satisfied both curve and a constrained applied on ‘a’ and ‘b’.

X	Y
1	1
3	24
6	1
8	30
14	3
24	27
25	22
32	17
33	29

Table 7.1 lookup table

Then generate private and public key pair for user A and user B

Let private key of user A is PR_A=13, and

Private Key of user B is PR_B =17 and public keys of user A and B are a multiplication of private keys and base point of curve.

Base point B_P= (1, 1).

And take another point that is affine point P_M (6, 1).

Now cipher text mean unreadable form of data C_F =(PR_A*B_P,P_{ML}+PR_A*PU_B)

7.1 Encryption

Takes an image and read each pixel value and transform scalar value of pixel into coordinate form by performing multiplication of pixel value with affine point.

Fig 7.1 original image

S_{ML}=a*P_M

Where a is pixel value of an image

And P_M is affine point

First takes binary value of pixels then using iteration method perform multiplication operation. Let pixel value i.e ‘a’ is 4 then Binary value of ‘a’ is 100.

Only 2 iterations is needed to perform multiplication operation and binary representation of pixel value contain 1, 0, 0 so perform only doubling operation two times and result is

4(6, 1) = (7.395623e+002, -2.011233e+004)

Then $PR_A * PU_B$ where PU_B is public key of user B it’s a multiplication of private key of user B with base point

$PU_B = PR_B * B_P = 17\ (1, 1) = (5.482019e+000, -1.300886e+001)$

$PU_B = (5.5, -13.00)$ (round off)

$PR_A * PU_B = 13*(5.5, -13.0) = (1.427888e+001, 5.407907e+001)$

$PR_A * PU_B = (14.2788, 54.0790)$ (round off)

$S_{ML} + PR_A * PU_B = (1.926959e+001, 8.468695e+001)$

$S_{ML} + PR_A * PU_B = (19.2696, 84.6869)$ (round off)

Encrypted Data-

$C_T = ((PR_A * B_P), (S_{ML} + PR_A * PU_B))$

$C_T = ((0.9867, -0.9734), (19.2696, 84.6869))$

Fig 7.2 encrypted image

7.2 Decryption At receiver side-

At receiver side, receiver receives cipher text that has two coordinate values first one is (PR_A *B_P) and another one is (S_{ML}+PR_A* PU_B) then multiply first value with private key of user B

PR_A *B_P *PR_B= (1.67234e+004,-6.85974e+007)

PR_A *B_P *PR_B =(16.7234e+003,-68.5974e+006)

Subtract this value from second coordinate value then resultant is an encrypted value of pixel so

(s1, s2) = (1.817513e+005, -7.748475e+007)

Now by the **Discrete Logarithmic Problem** get the original pixel value

Fig 7.3 decrypted image

Conclusion is that results are not clear and not exactly same as supplied in original because if looked on encrypted points then analyzed that they are not integer they are real numbers but after that approximate same result is obtained.

CHAPTER 8
CONCLUSION AND FUTURE WORK

From starting to last looking that ECC is successfully implemented on an image till date this is the latest and with less overhead and shorter key public key cryptography scheme which reduces system's effort in encryption.

This thesis summarizes that what is elliptic curve cryptography and what are the operations involved in the Encryption and decryption and analyze the results on an image. And also analyzed the look up table which contains the base point and all other affine points and more about elliptic curve, further discussed about what are type of attacks are possible in very much brief on ECC cryptography. Their countermeasures are not discussed so much because that is not our concern. Applications Elliptic curve cryptography in the real world and also in constrained environment, in some area this cryptography technique cannot be used. In the next generation ECC can be applied to IPV6 and now it is using the IPV4 protocol so in the new era ECC is build up a secure and effective protocol for IPV6.

ECC is applied on an image but it is not able to obtain exactly same result because in this thesis modular ECC is applied to get the result rather than Binary ECC. Why modular ECC is applied rather than binary ECC? Because modular ECC is based on modular calculus in which a mod value of the equation is determine. So the pixel values become closer as most possible so image display is a little bit different from original image but this is not our concern.

It has proven that ECC is a successful Public key generation and encryption technique. In this age when people is becoming more dependent on internet for communication and transferring files over public network such as internet then a such technology is necessary which is secure and unbreakable from network attacks i.e. ECC. In future this is a very wide area for research because many secrets of ECC can be disclosed that

are unrevealed still and that day is not so far on which ECC replaces RSA and other public key encryption techniques.

REFERENCES

[1] N. Kolbitz, Elliptic Curve Cryptosystems, *Mathematics of Computation, vol.48, 1987, pp.203-209.*

[2] William Stallings, CryptoGraphy and network Security, printice Hall, 5th edition.

[3] Standard specifications for public key cryptography, *IEEE standard, pl363, 2000*.

[4] Adam J. Albirt, "Understanding & Applied Cryptography and Data Security" CRC press,Pearson. And Internet sources URL: en.wikipedia.org

[5] Certicom, standards for Efficient Elliptic curve cryptography, SEC 1: Elliptic Curve Cryptography, Verson 2.0, May 21, 2009, Available at http://www.secg.org/download/aid-385/sec1_final.pdf

[6] Certicom website http://www.certicom.com/index.php?action=ecc_tutorial,home

[7] Anoop Ms "Elliptic Curve Cryptography-An implementation Guide" and from URL www.dkrypt.com.

[8] Alessandro Cilardo, Luigi coppolino, Nicola Mazocca, and Luigi Roman, "Elliptical Curve Cryptography Engineering", *Proceedings of the IEEE, vol. 94, no 9, pp. 395-406, Feb. 2006* .

[9] P. C. Kocher, "Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems," in *Advances in Cryptology—CRYPTO'96*, N. Koblitz, Ed.. Heidelberg, Germany: Springer-Verlag, 1996, vol. 1109, Lecture Notes in Computer Science, pp. 104–113.

[10] Internet source www.ece.wpi.edu

[11] K. Gandolfi, C. Mourtel, and F. Olivier, “Electromagnetic analysis: Concrete results,” in *Cryptographic Hardware and Embedded Systems—CHES 2001*, Ç. K. Koç, D. Naccache, and C. Paar, Eds. Heidelberg, Germany: Springer-Verlag, 2001, vol. 2162, Lecture Notes in Computer Science, pp. 251–261.

[12] I. Biehl, B. Meyer, and V. Muller, “Differential fault attacks on elliptic curve cryptosystems,” in *Advances in Cryptology— CRYPTO 2000*, M. Bellare, Ed. Heidelberg, Germany: Springer-Verlag, 2000, vol. 1880, Lecture Notes in Computer Science, pp. 131–146.

[13] Clare-Marie Karat, Jan Blom, John Karat, “"Introduction and Overview", Springer Professional Computing , 2004

[14] Craig Smith, “Basic Cryptanalysis Techniques”, November 17th, 2001website http://www.sans.org/reading_room/whitepapers/vpns/basic-cryptanalysis-techniques_752

[15] Junfeng Fa, Xu Guo, Elke De Mulder, Patrick Schaumont, Bart Preneel and Ingrid Verbauwhed *†2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* 978-1-4244-7812-5/10/$26.00_c 2010 IEEE 76-87.

[16] S.Maria Celestin Vigila, K.Muneeswaran, “Implementation of Text Based CryptoSystem using Eliptic Curve Cryptography”, 978-1-4244-4787-9/09/$25.00 ©2009 IEEE, ICAC 2009, 82-85.

[17] Po-Hsian,Huang, Ching-Wei, Chen “A study of the Elliptic curve cryptography applies to the Next Generation Protocol” 0-7803-8506-3/02/$17.00 © 2004 IEEE.

[18] Megha kolheker, Anita Jadhav, ”Implementation of Elliptic Curve Cryptography on text and image ” in International Journal of Enterprise Computing and Business Systems ISSN (Online) : 2230-8849 2011 url ijecbs.com

[19] Randhir Kumar and Akash Anil, "Implementation of Elliptical Curve Cryptography" in IJCSI International Journal of Computer Science Issues, Vol. 8,: 1694-0814 2011

[20] From internet source URL: www.danpritchard.com

[21] Manindra Agrawal, Neeraj Kayal, Nitin Saxena, "Prime is in P" Indian Institute of Technology URL: www.cse.iitk.ac.in/users/manindra/algebra/**primality**_v6.pdf

[22] Russell, Matthew."CRAN - CRyptANalysis toolkit". 21 Aug 2001.

URL: http://crank.sourceforge.net/about.html (24 Nov 2001).

[23] Tom Veerman, "Elliptic Curve Cryptography", June 21, 2010.

URL: http://www.science.uva.nl/onderwijs/thesis/centraal/files/f444779580.pdf

[24] Integer Factorization from Wikipedia

URL: http://en.wikipedia.org/wiki/Integer_factorization

[26] Types of Cryptanalysis or types of cryptographic attacks by Eric Conrad from

URL: http://www.giac.org/cissp-papers/57.pdf

[27] Shivangi Goyal," A Survey on the Applications of Cryptography", International Journal of Science and Technology Volume 1 No. 3, March, 2012

[28] Diffie Hellman Key Exchange from Wikipedia

URL: http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

[29] Diffie Hellman Problem from Wikipedia

URL: http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_problem

[30] Diffie Hellman Key Exchange from

URL: http://www.math.ucla.edu/~baker/40/handouts/rev_DH/node1.html

[31] ElGamal Cryptosystem from Networks data acquisition for the enterprise

URL: http://x5.net/faqs/crypto/q29.html

[32] ElGamal Encryption From Wikipedia

URL: http://en.wikipedia.org/wiki/ElGamal_encryption

[33] Professor Dr. D. J. Guan, "Pollard's Algorithm for Discrete Logarithm Problem" April 12, 2013 URL: http://guan.cse.nsysu.edu.tw/note/pollard.pdf

[34] Andreas Stein and Edlyn Teske, " Optimized Baby Step-Giant Step Methods" University of Wyoming, Department of Mathematics, P.O. Box 3036, 1000 E November 22, 2004 URL: https://www.math.uwaterloo.ca/~eteske/teske/St-jrm.pdf

[35] Public key Cryptography from Wikipedia

URL: http://en.wikipedia.org/wiki/Public-key_cryptography

[36] Elliptic Curve from Wikipedia

URL: http://en.wikipedia.org/wiki/Elliptic_curve or from URL en.wikipedia.org

[37] Elisabeth Oswald," Introduction to Elliptic Curve Cryptography" Institute for Applied Information Processing and Communication July 29, 2005

[38] Schoof's algorithm from Wikipedia

URL: http://en.wikipedia.org/wiki/Schoof's_algorithm

[39] Tetsuya Izu, Jun Kagure, Masayuki Noro and Kazuhiro Yokayama," Efficient Implementation of Scoof's algorithm" FUJITSU LABORATORIES LTD 211-8588 Japan

[40] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, Florida, USA, 1997.

[41] A. Woodbury, D. V. Bailey, and C. Paar. Elliptic Curve Cryptography on Smart Cards Without coprocessors. In IFIP CARDIS 2000, Fourth Smart Card Research and Advanced Application Conference, Bristol, UK, September 20–22 2000. Kluwer.

[42] From internet source www.infosecwriters.com