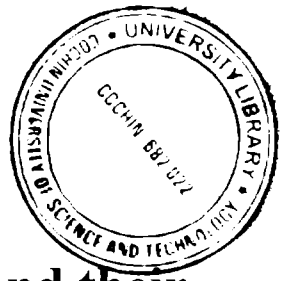


Algebraic Geometric Codes and their Relation to Cryptography using Elliptic Curves

Thesis Submitted to

Cochin University of Science and Technology

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Manju C

Under the guidance of

Dr. K.V Pramod

Department of Computer Applications

Cochin University of Science and Technology

Cochin – 682022

Kerala

April 2010

Algebraic Geometric Codes and their Relation to Cryptography using Elliptic Curves

Ph.D thesis in the field of Coding theory and Cryptography

Author

Manju C
Research Fellow
Department of Computer Application
Cochin University of Science and Technology
Cochin -682 022, Kerala, India
Email: manjuc76@gmail.com

Supervising Guide

Dr. K.V Pramod
Head of the Department
Department of Computer Application
Cochin University of Science and Technology
Cochin -682 022, Kerala, India
Email : pramod_k_v@cusat.ac.in

Cochin University of Science and Technology
Cochin – 682 022

Front Cover: An elliptic curve
April 2010

Dr. K.V Pramod
Head of the Department
Department of Computer Applications
Cochin University of Science and Technology
Cochin-682 022

Certificate

Certified that the work presented in this thesis entitled **"Algebraic Geometric Codes and their relation to Cryptography using Elliptic Curves"** is based on authentic record of research done by Ms. Manju C under my guidance and supervision at the Department of Computer Applications, Cochin University of Science and Technology, Cochin – 682 022 and has not been included in any other thesis for the award of any degree.

Cochin-22
Date : 19th April 2010

Dr. K V Pramod
(supervising Guide)

Declaration

I hereby declare that the work presented in this thesis entitled **"Algebraic Geometric Codes and their relation to Cryptography using Elliptic Curves"** is based on the original research work done by me under the guidance and supervision of **Dr K V Pramod,** Head of the Department, Department of Computer Applications, Cochin University of Science and Technology, Cochin and has not been included in any other thesis submitted previously for the award of any degree.

Cochin 22
Date :19.04.2010

Manju C

Acknowledgements

Here I pay my obeisance to the God almighty to have placed so many wonderful and kind people around me who have helped me lot in my work.

I express my sincere gratitude to my Supervising Guide **Dr K V Pramod**, Senior Lecturer and Head of Department of Computer Applications, Cochin University of Science and Technology, Cochin, who guided me all through my work and for his patience, kind care and pleasing behavior. His inspiration and support extended to me during the entire work is remembered here with thanks.

With profound gratitude, I express my sincere thanks to **Dr C.E Veni Madhavan**, Professor , IISc, Bangalore for suggesting the topic to me.

My sincere thanks are due to **Shri A. Sreekumar, Dr B Kannan** and **Smt. S. Malathi**, and all other faculties of the Department of Computer application, CUSAT, Cochin, for their help and suggestions they provided whenever I needed them the most.

Further, I thank Smt **Sindhu P Menon**, staff members and Library of the Department of Computer Applications, CUSAT, Cochin, who supported me to a very great extent.

Special thanks are due to the Principal and my colleagues of Bharathidasan Government College for Women, Puducherry who extended whole hearted help to materialize my aim. May I remember the encouragement and suggestions given by my dear colleagues.

My Parents were always with me. They stood by me all though the work, ready to support physically and morally. I extend my heartfelt gratitude to them. My brother **C Renju** who is working with Hindustan Aeronautics Limited, Bangalore as Design Engineer, had helped me a lot in course of my work with his technical acumen. I thank him here. My sincere thanks to my husband for his incessant follow ups and encouragements to complete my work.

Last, but not the least, to my seven year old son **Govind,** though he never understood what I have been doing, he never troubled me at any point of time. This work is being dedicated to my loving son.

Preface

Cryptography is the science of security of transmitting messages from a sender to a receiver. The objective is to encrypt message in such a way that an eavesdropper will not be able to read it. A cryptosystem is a system of algorithms for encrypting and decrypting the data for the purpose of ensuring security. Public key cryptosystem such as RSA and DSS are traditionally being preferred for the purpose of secure communication through the channel. However elliptic curve cryptosystem have become a viable alternative since they provide greater security and also because of their usage of key of smaller length compared to other existing crypto systems. Elliptic curve cryptography is based on group of points on an elliptic curve over a finite field.

Whenever data is transmitted across a channel, errors are likely to creep in. Coding theory is a stream of science that deals with finding efficient methods to encode and decode data, so that any likely errors can be detected and corrected. There are several methods to achieve coding and decoding. One among them is Algebraic Geometric Codes that can be constructed from curves. Clauds Shannon's 1948 paper "A mathematical theory of communications" give birth to twin disciplines - Information theory and Coding theory. The basic goal is efficient and reliable communication in an uncooperative environment. To be efficient, the transfer of information must not require a prohibitive amount of time and effort. To be reliable, the received data stream must resemble the transmitted stream with the narrow tolerances.

The main objective of the thesis is to relate Algebraic Geometric Codes to cryptography using Elliptic Curves. This is done by generating algorithms. Algorithms have been implemented with the help of MATLAB and comparative analysis is done with the help of existing algorithms.

The objectives of the thesis include

1. To generate cryptographic algorithms.
2. To combine decoding capability of Algebraic Geometric code with the process of decryption.
3. To ensure reliability of message by error detection and correction.
4. To ensure authenticity by generation of digital signature algorithm.
5. Secret key can be shared between authorized users so as to ensure more security.

The scope of the thesis is wide spread on transmitting the data. The security of the message can be achieved by applying keys of smaller size. Reliable data can be achieved by decoding the data so as to trace the data for errors. There by combining reliability with security

This thesis is based on six chapters. The chapter wise description is as follows. First chapter deals with Algebraic Geometric which include discussions on finite field, basics of Coding theory, Elliptic curves and Algebraic Geometric code. The Chapter two deals with basics of Cryptography. In this, a study on various cryptosystems which includes RSA, Elgammal, Elliptic curve cryptosystem and also existing cryptosystem using Algebraic Geometric code is done. In Chapter 3 a discussion on secret sharing is done. Secret sharing is the process of sharing secret among authorized users and the authorized group can only reveal the secret.

The chapter 4 deals with theoretical aspects of the problem. Here we are finding how Algebraic Geometric code can be related to Cryptography. This is done by generating algorithms. Here two algorithms have been developed. First algorithm convert message into Algebraic Geometric code and

the code is encrypted using secret key generated with the help of elliptic curve. To this algorithm at the receiving end, after the decryption process, decoding is done to detect and correct errors. The second algorithm considers a message as points on curve and generates a Generator matrix .The encryption is done using key from elliptic curve. The message takes form of a repetition code and decoding can be done efficiently at the receiving end by using these repetition codes.

In this thesis a secret sharing system is also developed where secret key is generated by using parameters of elliptic curve and distributing it among users. At the receiving end by getting shares from a set of authorized users secret can be reconstructed.

Digital signature is the subset of electronic signature that makes use of the concept of cryptography. Digital signature tries to combine the signature in real world taking into account, the properties of the electronic world. The properties that are provided and assured by the use of digital signatures include authentication, integrity and Non-repudiation. A digital signature algorithm for the above mentioned cryptosystem is also developed in this chapter. The last section of the chapter deals with the security aspects of the system. Attacks are common to all crypto systems, whenever we develop a system we should take into consideration of all the possible attacks that it is prone to. Here various cases have been studied and parameters that should be chosen to make the system away from all attacks are considered.

The Chapter 5 deals with the implementation of the algorithms that are developed. Implementations and various comparative analysis of the systems have been done by taking into considering various fields, messages of various lengths, Space requirement and also with other existing algorithms. In this chapter a study on various curves such as elliptic curves, hyper elliptic curves,

super-singular curves, and koblitz curve is also done and their applicability in generation of code and their usage in Cryptography is discussed.

The chapter 6 contains a comparative analysis of the system with the existing systems (Mc-Eliece , ECC) is done and finally conclusion is given, In this way we are trying to find a relation between Algebraic Geometric code and Cryptography. While Cryptography provides a secure way of sending messages, Algebraic geometric code converts the information to be transmitted, to a code. This code can be utilized to detect or correct errors. By combining these two, we are imposing security and error detection or error correction over our messages that are being transmitted.

CONTENTS

Chapter 5

Implementation

Chapter 6

Concluding Remarks and Some outlooks

INTRODUCTION

Communication is the process of transmitting data across channel. Whenever data is transmitted across a channel, errors are likely to occur. Coding theory is a stream of science that deals with finding efficient ways to encode and decode data, so that any likely errors can be detected and corrected. There are many methods to achieve coding and decoding. One among them is Algebraic Geometric Codes that can be constructed from curves.

Cryptography is the science of security of transmitting messages from a sender to a receiver. The objective is to encrypt message in such a way that an eavesdropper would not be able to read it. A cryptosystem is a set of algorithms for encrypting and decrypting for the purpose of the process of encryption and decryption. Public key cryptosystem such as RSA and DSS are traditionally being preferred for the purpose of secure communication through the channel. However Elliptic Curve cryptosystem have become a viable alternative since they provide greater security and also because of their usage of key of smaller length compared to other existing crypto systems. Elliptic curve cryptography is based on group of points on an elliptic curve over a finite field.

This thesis deals with Algebraic Geometric codes and their relation to Cryptography using elliptic curves. Here Goppa codes are used and the curves used are elliptic curve over a finite field. We are relating Algebraic Geometric code to Cryptography by developing a cryptographic algorithm, which includes the process of encryption and decryption of messages. We are making use of fundamental properties of Elliptic curve cryptography for generating the algorithm and is used here to relate both.

Concept of secret sharing is applied to the algorithm. Secret sharing is a scientific method for dividing key information into several places and

distributes it among the group of participants. Here we are making use of Shamir secret sharing schemes. In this method we are encrypting information using a secret key where as during the process of decryption, secret can be reconstructed by the shares given by the different participants.

Errors are likely to occur during the process of communications. We can decode an Algebraic Geometric code for the process of error detection and correction. Decoding methods can be applied to this Cryptographic algorithm to find whether any errors had occurred during the process of transmission. Various decoding methods are available that, can be applied to the algorithm to find whether errors have occurred in the information we communicated and can correct it.

In this way we are trying to find a relation between Algebraic Geometric codes to Cryptography. Cryptography provides a secure way of sending messages, while Algebraic Geometric code converts the information to be transmitted, to a code. This code can be decoded to detect or correct errors. By combining these two we are imposing security and error correction or detection over our message that are being transmitted.

In this thesis an algorithm is developed and implementation of the method is done by using MATLAB. A comparative study of the algorithm developed is done with existing public key crypto system, Elliptic Curve crypto system, and cryptosystem using Algebraic Geometric code. It is done to prove the efficiency of the system.

Attacks are common to all crypto systems whenever we develop a system. We should take into consideration of all the possible attacks that is prone to it. In this method various possible attacks is considered and a study is made regarding it.

Chapter 1
Algebraic Geometry

Contents

1.1 Finite field

A finite field is a field with finite number of elements. The order of finite field is the number of elements in the field. The order is always a prime or power of prime. Finite field is also called Galois field. Finite field is important in number theory, algebraic geometry, Galois Theory, Cryptography and Coding theory. The finite field is classified as follows [1]

- The order or number of elements of field is of the form p^n, where p is a prime number called the characteristic of the field and n is a positive integer.
- For every prime number p and positive integer n, there exists a finite field with p^n elements.
- Any two finite fields with same number of elements are isomorphic.

Notation for the finite field is F_{p^n} It can also be represented as GF (p^n) where GF stands for Galois field. The finite field GF (2) consists of elements 0 and 1.

1.1.1Construction of finite fields

To construct GF (p^n), first we have to find an irreducible polynomial or minimal g.

Definition 1.1(Irreducible polynomial [2]): A polynomial g ε F[x] is said to be irreducible over a finite field F if g has a positive degree and g = b.c with b, c ε F[x] implies that either b or c is a constant polynomial.

Definition 1.2 (Minimal Polynomial [2]): If θ ε F is an algebraic field over k, Then the uniquely determined monic polynomial g ε k [x] generating a sub string {f ε k[x], f (θ) = 0} of k[x] is called a minimal polynomial or irreducible polynomial of θ over k.

Properties of a Minimal polynomial include [2].

1. g is irreducible in k[x].
2. For f ε k[x], f (θ) = 0, if and only if g divides f.
3. g is monic polynomial in k[x] of least degree having θ as a root.

The polynomial g is of degree n with coefficient in z_p (for any prime p, z_p is the ring of integers) modulo p is field.

1.1.2Properties of GF (p^n) [3]

A Galois field will have following properties

i. It can be shown that for each positive integer n, there exists an irreducible polynomial of degree n over GF (p) for any p.
ii. It can be shown that for each divisor m of n, GF (p^n) has a unique sub field of order p^m moreover these are the sub fields of GF (p^n).

Theorem 1.1[3] Let θ ε F be algebraic field of degree n over K and let g be the minimal polynomial of θ over K. Then

1. K (θ) is isomorphic to K[x]/g.
2. | K (θ)| / |K| = n and {1, θ … , θ^{n-1}} is a basis of K (θ) over K.
3. Every α ε K (θ) is algebraic over K and its degree over K is a divisor of n.

Theorem 1.2: (Existence of Finite Field [3]) For every prime p and every positive integer there exists a finite field with p^n elements.

Theorem 1.3[3]: For every finite field F_q the multiplicative group F^*_q of non zero elements of F_q is cyclic.

Definition 1.3: (Primitive element [3]) A generator of the cyclic group F^*_q is called the primitive element of F_q.

Theorem 1.4[3] Let F_q be a finite field and F_r be a finite extension field , then F_r is a simple algebraic extension of F_q and every primitive element of F_r can serve as a defining element of F_r over F_q.

Theorem 1.5[3] If F is an irreducible polynomial in $F_q[x]$ of degree m, then F has a root α in F_q^m . Furthermore all roots of F are simple and are given by the distinct n elements $\alpha, \alpha^q, \alpha^{q2}, \ldots\ldots, \alpha^{qm-1}$ of F_q^m .

1.1.3 Finite field arithmetic

Arithmetic in finite field is different from standard integer arithmetic. There is limited number of elements in finite field and for all operations performed in the finite field, the result will be elements in the finite field. While each element in the finite field itself is not infinite, there are infinitely many different finite fields. Their number is necessarily of the form p^n, where p is a prime number and n is a positive integer. One can perform operations such as addition, multiplication, and subtraction using the usual operations on integers followed by modulo p.

E.g. GF (5)→4+5 = 9 is reduced to 4.

E.g. [3]: To represent the elements of F_q, Let F_q is regarded as a simple algebraic extension of F_3 of degree 2. This is obtained by conjunction of root of an irreducible quadratic polynomial over F_3 say $F(x) = x^2+1$ in $F_3[x]$. Thus $F(\alpha) = \alpha^2 + 1 = 0$ in F_9 and nine elements of F_9 is given by $\{0,1,2, \alpha, 1+\alpha, 2+\alpha, 2\alpha, 1+2\alpha, 2+2\alpha\}$. Here $a_0 + a_1\alpha$ form primitive element.

a. Addition and Subtraction

Addition and Subtraction of two finite fields are implemented in the intuitive way of adding and subtracting the coefficients and performing the modular reduction by subtracting or adding p until the resulting coefficient is non-negative and less than p.

b. Multiplication

Multiplication is done in two stages: Multiply two polynomials, A (x) and B (x), using ordinary polynomial multiplication to form an intermediate product c^{l} (x) to produce the result c (x).

c. Polynomial Exponentiation

Although raising a polynomial A (x) to the n^{th} power can be obtained by multiplying A (x) n times which is O (n), this is very difficult for large n. Thus repeated squaring can be used, which can be obtained with complexity O (log n).

1.2 Coding Theory

Clauds Shannon 's 1948 paper 'A mathematical theory of communications' give birth to twin disciplines Information theory and Coding theory. The basic goals of these disciplines are efficient and reliable communication in an uncooperative environment. To be efficient, the transfer of information must not require a prohibitive amount of time and effort. To be reliable, the received data stream must resemble the transmitted stream with narrow tolerances. [3]

1.2.1 General Communication System

Let us review the simplest communication scenario (The point – to-point communication).Shannon model of communication is as follows.

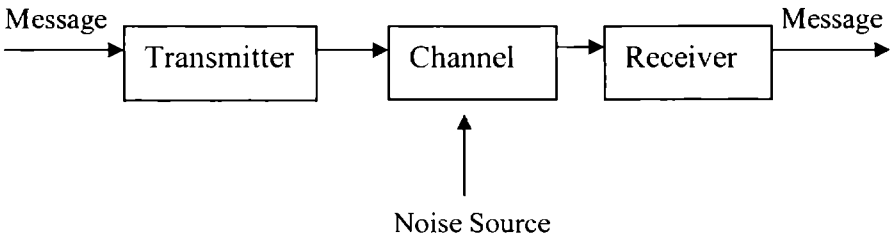


Fig 1.1 Model of communication process

A source (emitting speech, audio, data etc) transmits via a noisy channel (e.g.: phone line, optical link, wireless storage medium) to a destination. But we are interested in a reliable transmission i.e. we would like to recreate the transmitted information with as little distortion as possible as transmitted. A more specific model of communication can be as follows

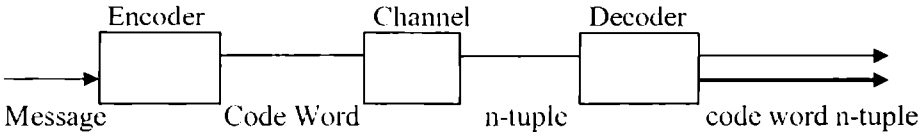


Fig 1.2 Reliable Model of communication process

Message is passed through an encoder, which encodes the message. By encoding; k-tuple message is converted into a code word of n-tuple. This code word is passed through a channel. A channel is a discrete memory less channel. The channel is discrete because we shall only consider finite alphabets. It is memory less, in the sense, error in one symbol will not affect the reliability of the neighboring symbols. The decoder receives from the channel an n-tuple of symbols. Then the decoder decodes the information to get the transmitted image. Decoding involves the process of error detection and correction also.

The aim of coding is to provide secure transmission of message, in the sense that errors occurred during the transmission can be corrected. In the coding theory, basic thing is the creation of code words. The code word should be created in such a way that it is possible for

- Fast encoding of information.
- Easy transmission of encoded messages.
- Fast decoding of received messages.
- Correction of errors introduced in the channel.
- Maximum transfer of information per unit time.

1.2.2 Codes and Types of codes

A code C over an alphabet A is simply a subset of $A^n = A \times A \times .....x$ A (n copies). Elements of a code are called code words and the length of the code is n, where A is a finite field F_q The dimension of a linear code C is defined as a vector space over F_q.

Important parameters of code include

1. Information rate: It is the number that is designated to measure the proportion of each code word that is carrying the message and it is given by k/n where k is the dimension of code and n is the length of the code.
2. Relative distance: It is the number of positions where two code words disagree.

Different types of codes include:

a. Linear Code

A code C is a linear code, if v + w is a word in C so that v and w are in C. i.e. linear code is a code which is closed under addition of words.

For e.g.: c = {000,111} is a linear code since

000+000=000

111+111=000

111+000=111

000+111=111

The distance of a linear code is equal to minimum weight of any non-zero code word. The parameter of a linear code can be defined as (n, k, d) where n is the length of the code, k is the dimension of the code and d is the distance of the code.

If C is a linear code of length n and dimension k, then any matrix whose rows form the basis for C is called generator matrix for C. Generator matrix for a linear code must have k rows and n columns and rank K.

Theorem 1.6[3] Generator Matrix:- A matrix G is generator matrix for some linear code C if and only if rank G is equals to number of rows of G.

Let C be a linear code of length n and dimension k, if G is a generator for C and if u is a word of length k written as a row vector then

$$V = uG \tag{1.1}$$

is a word in C.

There is another matrix associated with a code and closely connected with a generator matrix. This matrix is called a parity check matrix and this matrix plays an important role in decoding of code, which is used for the purpose of error correction and detection.

A matrix H is called a parity check matrix for a linear code if the columns of H form a basis for the dual code $C^{\perp}$. If C has a length n and

dimension k, then any parity check matrix of C must have n rows and n-k columns and rank n-k.

Theorem 1.7[3]: A matrix H is a parity check matrix for some linear code C if and only if the columns of H are linearly independent.

Theorem 1.8[3]: If H is a parity check matrix for a linear code C of length n then C consists precisely of all code words V in K such that

$$VH=0 \qquad (1.2)$$

These results can be used for error detection and correction.

Theorem 1.9[3] Matrix G and H are generating and parity check matrices, respectively for some linear code if and only if

i. The rows of G are linearly independent.
ii. The columns of H are linearly independent.
iii. Number of rows of G plus the number of columns of H equals the number of columns of G, which is equal to number of rows of H.
iv. GH=0.

b. Hamming Code

A code of length $n = 2^r-1$, $r \geq 2$ having parity check matrix H whose rows consists of all non-zero vectors of length r is called a hamming code of length 2^r-1. Parity check matrix H for a hamming code C contains all r rows of weight r and r columns of H are linear independent. Thus hamming code has dimension $2^r - 1 - r$ and contains 2^{2^r-1-r} code words. Hamming codes are perfect error correcting codes.

c. Reed-Muller codes

This is another important class of codes, which includes the extended Hamming code. The r^{th} order Reed-Muller code of length 2^m is denoted by RM(r, m), $0 \leq r \leq m$. We present a recursive definition of these codes

1. R (0, m) = {0, 0, 0, 1, 1 1, 1...1}
2. RM (m, m) = K^{2^r}
3. RM(r, m) = {(x, x + y)| x ε RM(r, m-1), y ε RM (r-1, m-1)}; 0 < r < m

Generator matrix of RM(r, m) is defined by

$$G(r, m) = \begin{bmatrix} G(r, m-1) & G(r, m-1) \\ 0 & G(r-1, m-1) \end{bmatrix}$$

Theorem 1.10 [3]: The r^{th} order Reed-Muller code RM(r, m) defined above have the following properties

1. Length $n = 2^m$
2. Distance $d = 2^{m-r}$
3. Dimension $k = \sum {}^{r}{}_{m}(i)$
4. RM(r-1, m) is contained in RM(r, m)
5. Dual code RM(m- 1-r,m) , r< m

d. BCH codes

An Important class of multiple-error correcting code is the class of Bose Chaudhari-Hocquingham codes or BCH codes. They are important because of two reasons. Firstly they admit easy decoding scheme and secondly the class of BCH code is quite extensive.

e. Reed-Solomon code

Reed Solomon code is a linear systematic block code based on finite field theory. The basic building block of Reed-Solomon codes is a symbol composed of m binary bits, where m can be any natural numbers greater than or equal to 2. For a given length m, the length of all the Reed-Solomon codes composed of m bit symbols is 2^m-1. Reed-Solomon code is a special case of BCH code. An efficient algorithm for BCH code was discovered in 1968. We can apply same thing to reed-Solomon code also. An alternate definition of Reed-Solomon code is as follows

Definition 1.4 [3]: Given a finite field F of size q, let n = q-1 and let α be a primitive n^{th} root of unity in F. Also, let $1 \leq K \leq n$ then, the Reed-Solomon code for these parameters has a code word $(f_0, f_1 \ldots f_{n-1})$. If and only if α, α^1, $\alpha^2 \ldots\ldots \alpha^{n-k}$ are root of polynomial $p(x) = f_0 + f_1(x) + \ldots\ldots\ldots + f_{n-1}x^{n-1}$.

With this definition, it is immediately seen that a Reed-Solomon code is a polynomial code. Let g (x) be a generator polynomial, which is minimal, roots $\alpha, \alpha^2, \alpha^3 \ldots\ldots \alpha^{n-k}$ and the code words are exactly the polynomials that are divisible by g (x).

Let $p(x) = V_0 + V_1x + V_2x^2 + \ldots + V_{n-1}x^{n-1}$ and $q(x) = f_0(x) + f_1(x) + \ldots\ldots f_{n-1}(x^{n-1})$, using these facts we have

- $(f_0, f_1, \ldots\ldots f_n)$ is a code word of the Reed-Solomon code
- iff p(x) is of degree < k.
- iff $V_i = 0$ for i = k, . . ., n-1.
- iff $q(\alpha^i) = 0$ for i =1,....n-k.

We can say in Reed – Solomon code data encoded is visualized as a polynomial. The code relies on a theorem from algebra that states that any k distinct point uniquely determines a polynomial of degree at most k-1.

Next code that is important and used in our algorithm is algebraic geometric code. Algebraic geometric code is code based on Algebraic Curves. Here in our system we are making use of Elliptic curves. The detail of the code is given in section 1.4.

1.2.3 Bounds on Codes

Quality of codes is determined by two variants. They are transmission rate and relative distance. The transmission rate is given by R = k/n and relative distance given by δ = d/n, where n is the length of the code, k is the dimension and d is the minimum distance. The main aim of coding theory is to generate codes that optimize these parameters [4, 5, 6]. Quality of code can be defined using bounds on codes.

a) Singleton bound

Let C be a code over a finite field with dimension k, minimum distance d and length n then $d \leq n-k+1$. Any code having parameters, which meet singleton bond, is maximum separable code.

Definition 1.5[4] : Let q be a prime power and let n, d be positive integers with $d \leq n$, then the quantity $A_q(n, d)$ is defined as maximum value of M, such that there is a code over F_q of length n with M code words and minimum distance d. By singleton bound, we will have $A_q(n, d) \leq q^{n-d+1}$.

b) Plot kin bound

Here we will set $\theta = 1-1/q$, then $A_q(n, d) = 0$ if $d < \theta_n$ and $A_q(n, d) \leq d/d-\theta_n$, if $d > \theta_n$.

If we have a code that satisfied that the above condition, then we will say it is having Plotkin bound [4].

c) Gilbert – Vaishnamov bound

Here $A_q(n, d) \leq q^n / v_q(n,d-1)$

d) Asymptotic bounds

Since we are looking for codes, which have large dimension and large minimum distance with respect to n, it makes sense to normalize these parameters by dividing by n. Let C be a code over F_q of length n with q^k code words and minimum distance d. As specified, R and δ determines quality of code. Both R and δ should be between 0 and 1 and C is a good code if both R and δ is close to 1. Let q be a prime power and $\delta \varepsilon R$ with $0 \leq \delta \leq 1$ then

$\alpha_q(\delta) = Lt(\sup(1/n(\log_q(A_q(n, d)))))$.

We will see asymptotic version of Plotkin and Gilbert Varshnamov bounds. These bounds, give bounds on the value of $\alpha_q(\delta)$.

Asymptotic Plotkin bound [3, 4] is as follows

With $\theta = 1 - 1/q$ we have $\alpha_q(\delta) = 1 - \delta/\theta$, if $0 \leq \delta \leq \theta$

$$\alpha_q(\delta) = 0, \quad \text{if } 0 < \delta \leq 1.$$

In order to define Gilbert Varshnomov bound we have to specify Hilbert entropy on set $\theta = 1-1/q$ and define a function $H_q(x)$ on an interval $0 \leq x \leq 0$ by $x=0$.

$$Hq(x) = \begin{cases} 0 & \text{if } x = 0 \\ x \log_q(q-1) - x \log_q(x) - (1-x)\log_q(1-x) & \text{if } 0 \leq x \leq q \end{cases}$$

The function $H_q(x)$ is called Hilbert entropy function.

Asymptotic Gilbert – Varshamov bound [4] can be defined as follows find any δ with $0 \leq \delta \leq q$,we have $\alpha_q(\delta) \geq 1\text{-}Hq(\delta)$. The Fig 1.3 is a graph for Plotkin bound and Gilbert Varshamov bound.

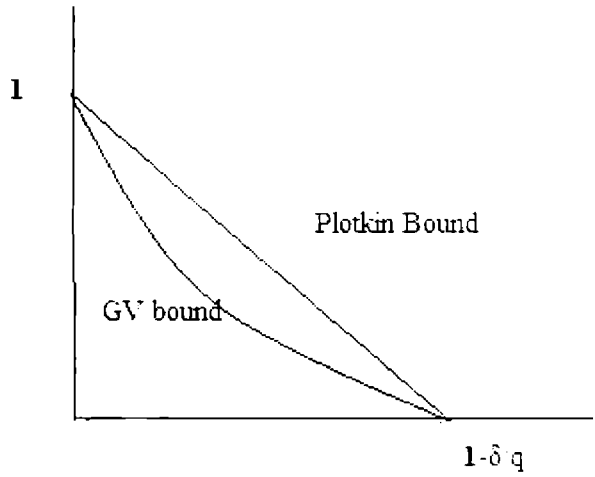


Fig 1.3 Plotkin and GV Bound

1.3. Elliptic curves

1.3.1 Introduction

Elliptic curves are becoming more and important, not only as cryptographic applications but as important in mathematical theory. Elliptic curve originated from early mathematicians trying to find rational solutions to the cubic equations. Curves can be defined over affine plane, by adding points at infinity to affine plane to produce a projective plane and projective plane can be defined as follows

1. Affine Coordinates: Co-ordinates of the form f (x, y) = 0 where f is a non-zero polynomial.

2. Projective coordinates [8]: coordinates of the form f (x, y, z) =0 where f is a non – zero polynomial of some degree d and projective rational solutions are (x, y, z) and (λx, λy, λz) for λ = 0.We can define elliptic curve over a finite field F_p as follows

Definition 1.5[7] E: $y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6$ (1.3)

Where $a_1, a_2 \ldots a_6 \varepsilon F_p$ and $\Delta \neq 0$, Δ is the discriminant of E and is defined as follows

1. $\Delta = -d_2^2 - 8d_4^3 - 27 d_6^2 + 9 d_2 d_4 d_6$

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

3. $d_4 = 2 a_4 + 4 a_2$

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

5. $d_8 = a_1^2 a_6 + 4 a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2$

Let L be the extension field of F_q then the set of rational points on E is

$E(L)=\{(x, y) \varepsilon L \times L: y^2 + a_1 x y + a_3 y - x^3 - a_2 x^2 - a_4 x - a_6 =0\} \cap \text{union}\{\infty\}$ (1.4)

Where ∞ is point at infinity and certain properties of the curve are as follows.

1) Equation –(1.3) is a Weierstrass equation.

2) Condition $\Delta \neq 0$ ensures that the elliptic curve is smooth, that is, there are no points at which curve have one or more tangent lines. The point ∞ is the only point on the line at infinity that satisfies the projective form of Weierstrass equation.

3) The L-rational function points on E are the points (x, y) that satisfy the equation of the curve and whose coordinates x and y belongs to L.

The point at infinity is considered as L rational point for all extension fields of L on Fp. We can transform the elliptic curve Equation (1.3) to

$$y^2 = x^3 + ax + b \qquad (1.5)$$

where a and b ε F_p, F_p is finite field of size p. Such a curve is said to be super singular and has discriminant $\Delta = -a^3$. In this thesis we will be using elliptic curve of equation (1.5).Consider an elliptic curve $y^2 = x^3 + 6x + 5$ over field f_8. The curve can be represented as follows as in Fig 1.5.

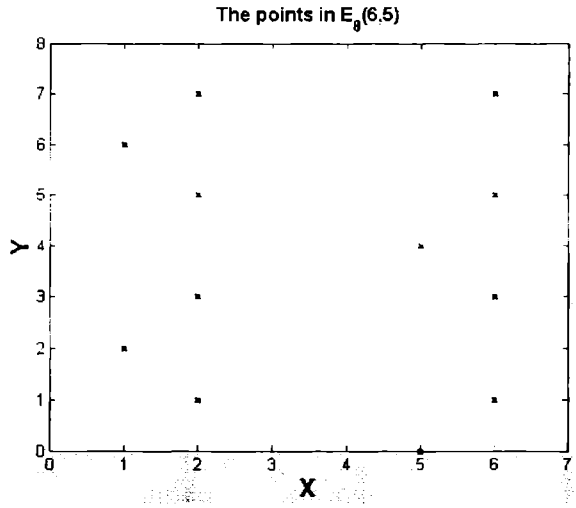


Fig 1.4 Elliptic curve point represent representation $E_8(6,5)$

This is the representation of curve along with points using algorithm in MATLAB as specified in Appendix-A-II

1.3.2 Properties

Properties of an elliptic curve $y^2 = x^3 + ax + b$ [7] are

1. Identity: $P + \infty = \infty + P$ for all $P \varepsilon E (F_p)$.

2. Negatives: If P (x, y) ε E (F_p) then (x, y) + (x, -y) = ∞. The point (x, -y) is denoted as –p and is called negative of P.

1.3.3 Arithmetic of Elliptic curves

Arithmetic of elliptic curve includes all operations which can be done on points on a curve

i. **Point addition**

Let P (x_1, y_1) ε E (F_p) and Q (x_2, y_2) ε E (F_p) where P ≠ ± Q, then P +Q = (x_3, y_3) where

$$x_3 = (y_2-y_1/x_2-x_1)^2 - x_1 - x_2 \qquad (1.6)$$

$$y_3 = ((y_2-y_1/x_2-x_1)(x_1-x_3) - y_1) \qquad (1.7)$$

Addition [7, 8] can be done by chord and tangent rule .The sum R of two points P and Q is as follows. Draw a line through P_1 and P_2, this line intersect at 3 rd point. Then P_3 is the reflection of this point about the x-axis.

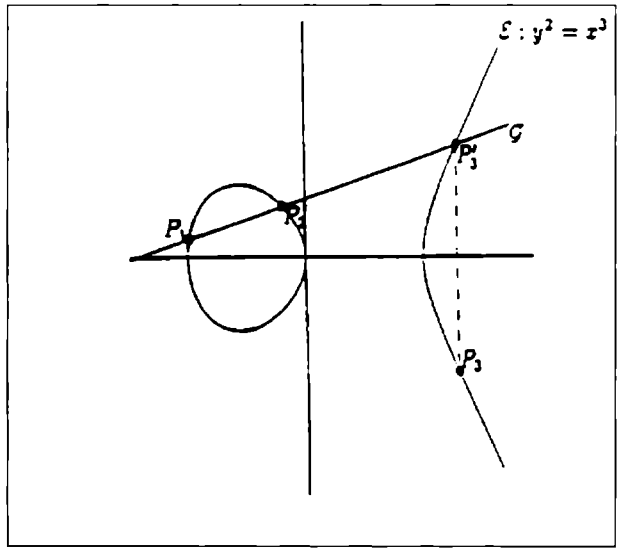


Figure:1. 5 Elliptic curve point addition

An example for elliptic curve addition is as follows. The curve equation is $y^2=x^3+ax+b$ with a = - 4, b = 4.To add two points, draw a line through them and reflect the third point, where this line intersects the curve, in the x-axis.

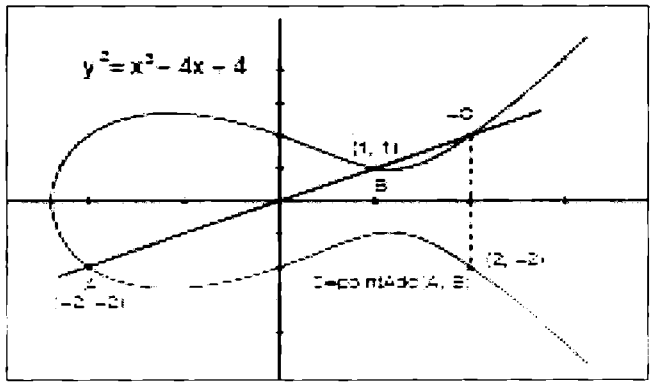


Fig 1.6[8] (Elliptic curve addition example)

The result of addition of points A (-2, -2) and B (1, 1) is C (2, -2).

ii. Point doubling

Elliptic curve doubling occurs when P = Q. Doubling a point is a process of computing P+P, P ε E. Let m be the slope of the curve,

If $y_1 \neq 0$ then $x_3 = m^2 - 2\,x_1$, $y_3 = m\,(x_1 - x_3) - y_1$. Here we take slope as $(3\,x_1^2 + a)/\,2\,y_1)$.

If $y_1 = 0$ then $P_1 + P_2 = \infty$. Also we can say $P + \infty = P$ for all points P on E.

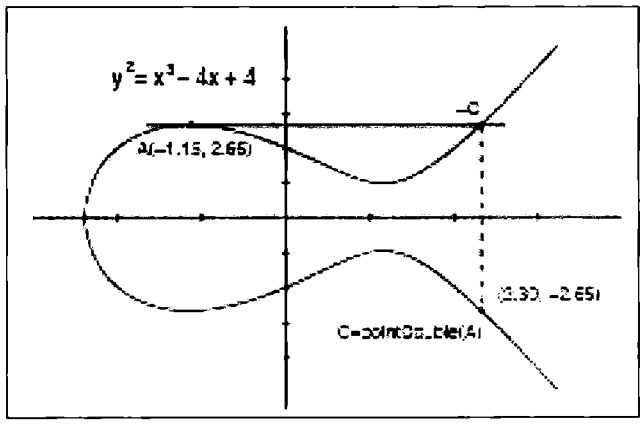


Figure 1. 7 [8]. Elliptic curve doubling

iii. Scalar multiplication

Scalar multiplication is a process of multiplying a scalar value with a point on curve. Let P be a point on curve, kP, that means scalar multiplication of an integer k with a point P. That is P + P + P+ . . . . . +P (k times). It is done by doubling and adding method. This property of elliptic curve is used in this thesis for implementation of cryptographic algorithm. Elliptic curve used in cryptography contains only finite number of points. The figure below shows scalar multiplication on elliptic curves.

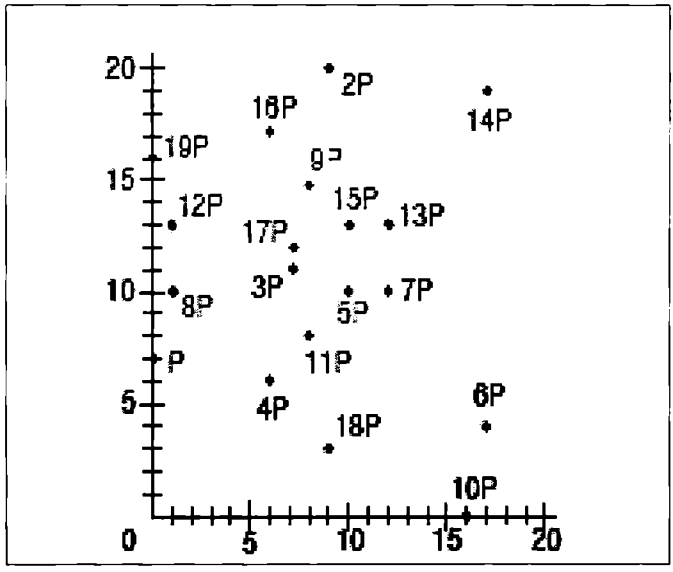


Figure 1.8 Scalar multiplication

iv. Point Subtraction

Point subtraction consists of point addition and point negation or point inverse. It is represented as P – Q. This can be evaluated by point addition of the point P and inverse Q (-Q). i.e. R = P + (-Q).

1.4. Algebraic Geometric Codes

Algebraic Geometric codes are codes defined over curves. Algebraic geometric code is defined by V.D Goppa [5, 9, 10]. The curve used in algebraic geometric code is defined over a finite field F_q. here we can make use of affine and projective variety of curve whose dimension is one. This curve is absolutely irreducible and non-singular, equations of curve should be polynomials with coefficients of F_q.

The Key aim of algebraic geometric code is to replace polynomial over a finite field by more general constructions. Goppa used language of algebraic curves to introduce codes. So we can call it as Algebraic Geometric codes. Before going into function and construction of algebraic geometric codes we will discuss certain factors used for describing the code. They include

1.4 .1 Divisors

A divisor[6] D on a curve X is a formal sum of form $D=\sum n_p P$ where $n_p \varepsilon Z$ and $n_p = 0$ for all but a finite number of points P on X. Divisors are often thought to be the key stone to understand how Algebraic Geometry is formed and its relationship to curve. To describe it more clearly, let C be a non-singular projective curve in P_k^2.The projective plane is over an algebraically closed field K. For each line L in P_k^2, we consider $L \cap C$, which is a finite set of points on C. If C is a curve of degree d and if we counts points with proper multiplicity then $L \cap C$ will contains exactly d points. So we can write $L \cap C = \sum n_p P$ where $P_i \varepsilon C$ are the points, n_i the multiplicity and this formal sum is a divisor on C. As L varies, we obtain a family of divisors on C parameterized by the set of all lines in P^2, which is a dual projective space $(P_k^2)^*$. We refer to this set of divisors as a linear system of divisors on C. If P is a point of C, the set of divisors in the linear system contains P. They correspond to the lines $L \varepsilon (P_k^2)^*$ passing through P and this set of lines determines P uniquely as a point in P_k^2.

Another important thing in the construction of Algebraic Geometric code is order function. The order is a generalization of the degree of a function as well as its zeroes. There are two candidates, the x-order and the y-order. Usually they are the same; however care must be taken to ensure their accuracy.

Definition 1.6[6]: Let X: f (x, y) = 0 be a curve and P (x=α, y=β) be a point on curve X with α and β ε F, Let g (x, y) ε F [X], then the largest power n for which there exists polynomials g^0 ε F [X] and h^0 (x, y) ε F [x, y] with h^0 (0, 0) ≠ 0 such that

$$g= ((x- \alpha)\, g^0(x- \alpha)/h^0(x- \alpha, y- \beta)) \bmod f$$

is called the x-order of g at P and denoted by $ord_{p.x}(g)$.The x-order can be defined using the notation $V_{p.x}(g/h)$ and is $V_{p.x}(g)$ - $V_{p.x}(h)$ and y order is defined analogously.

Let F be a field. A discrete valuation V on F is a function f to Z. It has the following properties

- V (a, b) =V (a) + V (b)
- V (a + b) ≥ min {V (a), V (b)}
- V (a) =1 for at least one a.

It is some times convenient to put V (0) = ∞, which preserves the axioms even when a=0 or b=0.

Proposition 1.1[8]: If V is a discrete valuation then

V (1) = 0 and if V (a) < V (b) then V (a + b) = V (a).

Assume further X is projective and f ε K $(X)^*$ then the following is equivalent

$Ord_y(f) \geq o$ for all y

$Ord_x(f) = o$ for all y . f ε K^* .

The properties of ord_y shown above allow us to define the divisor of a function. Let X be a variety and $f \varepsilon K(X)^*$ be a rational function (section 1.4.2) on X, Then divisor of f is

$$Div(f) = \sum_y ord_y(f)\, y \varepsilon Div(X).$$

A divisor is said to be principal if it is the divisor of a function. Two divisor D and D^1 are linearly equivalent ($D \sim D^1$), if the difference is a principal divisor. We can use notation (f) for the divisor of f. The divisor at poles and zeroes denoted by $(f)_0$ and $(f)_\infty$ respectively.

$$(f)_0 = \sum_{ord_y(f) \geq 0} ord_y(f) y.$$

$$(f)_\infty = \sum_{ord_y(f) < 0} - ord_y(f)\, y.$$

Thus we can say that divisor of a function is the difference between poles and zeroes.

1.4.2 Rational Functions

Let X is a curve defined by a field F. On the points of X, any two polynomials that differ by multiples of F have same value. So when we compare it with the curve they will be the same.

Definition 1.7 [6]: Rational function R as the ratio $f = (x, y, z)/B(x, y, z)$ of two homogeneous polynomials of the same degree up to factorization modulo $F(x, y, z)$.

A rational function f is defined at a point P, if there exists a representation $f = A/B$ such that $B(P) \neq 0$.

Another important thing we have to discuss before the construction and definition of algebraic geometric code is the space associated with the divisor. The space associated with the divisor can be called as linear space.

Let $D = \sum n_p P$, be a divisor and there are set of all functions satisfying $Vp(f) \geq -n_p$ at every point P, together with the zero function is called space associated to D and is denoted by L (D). For an effective divisor D, L (D) consists of rational functions and all its poles lie in the Supp (D) and the multiplicity of each of them is not greater than n_p . We can describe it with the help of a Lemma and a proof.

Lemma [1.1] Let $D \in D_x$ then

1. If D^1 is linearly equivalent to D, then L (D) is isomorphic to L (D^1) (as a vector space over K)
2. If deg (D) < 0 then L (D) = {0}
3. L (0) = K.

Proof: (1) As D and D^1 are equivalent there exists z c K (X) such that $D = D^1 + (z)$. Define a mapping Φ: L (D)→K (X), $x \mapsto xz$. Clearly Φ is k-linear and its image is contained in L (D^1): $Vp(xz) = Vp(x) + Vp(y) \geq -np + Vp(Z) = -np$ for every $P \in X$. More over Φ is bijective as ψ: L(D^1)→L (D), $x \mapsto xz^{-1}$ is an inverse of Φ.

(2)Assume that there exists $x \in L(D)$, $x \neq 0$, then $D^1 = D + (x)$ is effective and linearly equivalent to D. Hence, $0 \leq \deg(D^1) = \deg(D)$, which is a contradiction

3. Clearly K is contained in L (0). On the other hand,, each element in L (0) has no poles, therefore it is a constant.

1.4.3 Riemann –Roch Theorem.

It is one of the famous theorems in algebraic geometry. It deals with computation of l(D) ,the dimension of vector space L(D).Let X be a curve defined over a projective field and let d be the degree of X , g the genus of curve =(d-1)(d-2)/2. A canonical divisor w is also defined such that deg (w) = 2g-2 and l (w) = g.

Theorem 1. 11 [5] Given a divisor D, l(D) = deg (D) + 1- g + l(w-D), where w is any canonical divisor.

By making use of all the above discussed concepts of algebraic geometry we can define an algebraic geometric code by V.D Goppa as follows,

Let X be a curve, P be a set of points on the curve, D be the divisor then Algebraic Geometric code associated to (X, P, D) is

$$C(X,P,D):= ((f(p_1),f(p_2).....,f(p_n) \mid f \in L(D)\} \subset F_q^n \quad (1.8)$$

In other words, the algebraic geometric code C(X, P, D) is the image of the evaluation map

$$E:L(D)\rightarrow F_q^n$$

$$f\rightarrow ((f_1(p_1),f_2(p_2).....f_k(p_n)) \quad (1.9)$$

1.4.4 Construction of Algebraic Geometric Code

By making use of the definition described above we can construct a Goppa code as follows .Let X be a curve, P be a set of n points on the curve $\{P_1,P_2 \ldots P_n\}$ and divisor $D = P_1+P_2 \ldots P_n$ Let L(D) denote vector space for the curve X, length of the vector space l(D) as per Riemann Roch theorem is l(D) = n + g - 1. Let g =1, genus of an elliptic curve is one(in this thesis we are concentrating only on elliptic curve). For an elliptic curve d=3, and genus g is given by ((d-1)(d-2))/2. So g here is 1. Then l (D) = n = #p (number of points on the curve). A code is represented by (n, k, d) from where n is the number of elements, k is the dimension and d is the distance.

The dimension K is degD+1-g and minimum distance d > n-deg D (Thus we are mapping (X,P,D) to (n, k, d) curve). Let C=(X,P,D) be an algebraic geometric code and let $f_1, f_2 \ldots f_k$ be a basis for the vector space L(D) over finite field F_q under the conditions above dim C= K and geometric matrix is defined as

$$\begin{pmatrix} f_1(P_1) & \cdots & f_1(P_n) \\ \cdot & & \\ f_k(P_1) & \cdots & f_k(P_n) \end{pmatrix}$$

An example for an Algebraic Geometric code over Hermition curve is as follows. Let our curve be $F=V(x^3+y^2z+yz^2)$ over the field F_4. As this curve is smooth we can find genus of a curve by using Pluckers formula as $g = (3-1)(3-2)/2 = 1$. This curve contains 9 points.

$Q = (0 : 1 : 0)$

$P_1 = (0 : 0 : 1)$ $P_2 = (0 : 1 : 1)$ $P_3 = (1 : \alpha : 1)$

$P_4 = (1 : \alpha^2 : 1)$ $P_5 = (\alpha : \alpha : 1)$ $P_6 = (\alpha : \alpha^2 : 1)$

$P_6 = (\alpha : \alpha^2 : 1)$ $P_8 = (\alpha^2 : \alpha^2 : 1)$

Let D be the divisor of the sum of eight affine points, that is

$$D = P_1 + \ldots\ldots\ldots + P_8.$$

Let the code be C_L (D, 4Q). The 4 dimensional space L (4Q) is spanned by the following basis functions. The numbers in parentheses indicate the order of the pole at point Q.

$\Phi_1 = 1$ (0) $\Phi_2 = x/z$ (2) $\Phi_3 = y/z$ (3) $\Phi_4 = x^2/z^2$ (4)

with this information we can already give a generator matrix for the code C_L (D, 4Q) by evaluating $\Phi_1, \ldots\ldots, \Phi_4$ by evaluating the functions at points $P_1, \ldots\ldots P_8$.

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & \alpha & \alpha & \alpha^2 & \alpha^2 \\ 0 & 1 & \alpha & \alpha^2 & \alpha & \alpha^2 & \alpha & \alpha^2 \\ 0 & 0 & 1 & 1 & \alpha^2 & \alpha^2 & \alpha & \alpha^2 \end{pmatrix}$$

1.5 References

[1] Jacobson, Nathan, " Basic Algebra 1(second edition)" New York,W.H Freeman & Co, 1985, ISBN 978-0-7167-1480.

[2] R. Liedl, H.Niederreater, "Finite fields, Mathematics and its Applications", Vol 20, Cambridge University Press, 1984.

[3] D.R Hankerson, D. Hoffman, D.A Leonard, C.C Linder, T.T Phelps, C.A Rodger, J.R Wall, "The Coding theory and Cryptography The Essentials", 2004.

[4] F.J Mac Williams and N.J.A Sloane, "Theory of Error Correcting Codes", Elsevier publishers, New York, 1997.

[5] V.D Goppa, "Codes on algebraic curves", Sov. Math, Dokl, Pages 207-214, 1981.

[6] Hart Shorne, "Algebraic Geometry", 1977 ,Graduate text in mathematics, Vol 52, Springer Verlag, 1986..

[7] Hankerson, Menzes, Vanstone, "Guide to elliptic curve cryptography", Springer, CRC press, 2004.

[8] Shueling Chang, Hans Eberle, Vipul Gupta , "Elliptic Curve Cryptography-How it works " , Sun Microsystems.

[9] H. Stichtenoth, "Algebraic function fields and codes", Universitext, Springer- Verlag, Berlin-Heidelberg, 1993.

[10] J.H. Van Lint, "Introduction to Coding Theory", Grad. Texts in Math, Vol. 86, Springer-Verlag, New York-Heidelberg-Berlin, 1982.

.....ଌ.....

Chapter 2
Cryptography

Contents

2.1 Public key Cryptography

2.1.1 Introduction

The fundamental aim of cryptography has always been to provide secure communication over a channel. Cryptography includes broad range of science including mathematics, computer science, information theory and human psychology. Cryptographic systems can provide a number of services with application emphasizing the determination. These services are the building blocks of a secure system and is defined by industry [1] as follows

- Confidentiality-Concealment of data from all but authorized parties.
- User authentication-Assurance that the parties involved in a real time transaction.
- Data origin authentication: Assurance of the source of message
- Data integrity: Assurance that data has not been modified by unauthorized parties.
- Non-repudiation-The binding of an entity to a transaction in which it participates, so that the transaction cannot be later repudiated. That is, the receiver of a transaction is able to demonstrate to a neutral third party, that claimed sender did indeed send the transaction.

- Availability- A measure of ability of the system to function efficiently in providing the security.
- Data integrity and non-repudiation of the information transmitted can be achieved by digital signatures.

Cryptography is the process of converting ordinary plain text units called plain text messages into units of encrypted text called cipher text message units. The conversion process is done using a secret key.

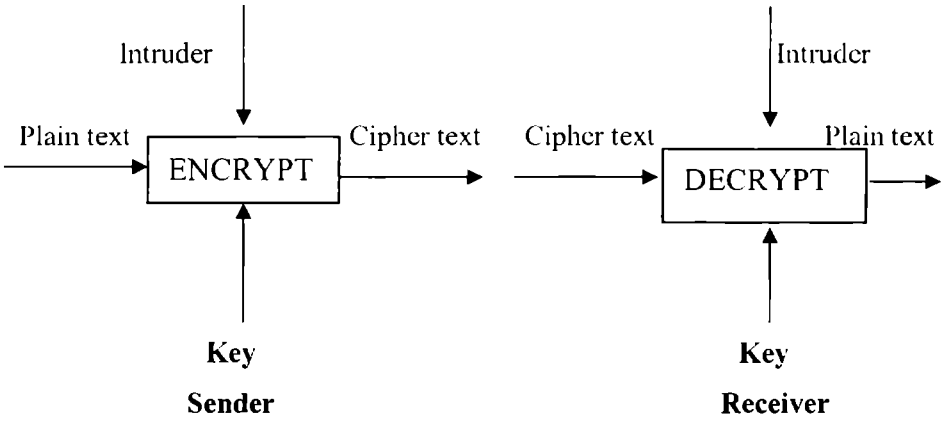


Fig 2.1 Process of Cryptography

After converting plain text into cipher text, sender transmits the message through communication channel. At the receiving end the cipher text is converted into plain text, using key along with the decryption algorithm.

There are two types of Cryptography: - Symmetric and Asymmetric cryptography. Asymmetric cryptography is useful for secure communications and they don't use single key, where as Symmetric cryptography uses single key. Asymmetric cryptosystems use two keys, in which one is a public key and the other is a private key. Public key is used for encryption and private key is used for the process of decryption. So we can call Asymmetric cryptosystem as a public key cryptosystem.

Major objective of public key encryption is to provide privacy and confidentiality. In public key encryption each entity A has a public key e and corresponding private key d. In a secure system, the task of computation of d from e is computationally infeasible. The public key defines the encryption transformation and private key d defines decryption transformation. So if any entity B wants to transmit a message M to A, it obtains A's public key e and uses encryption transformation to obtain cipher text. So cipher $C = E_e(m)$ and transmit C to A. At receiving end A decrypt the cipher text by decryption transformation to get original message m. $m=D_d(c)$. Public key is not secret, it is known to all and where as private key is secret.

In 1976, Diffie and Hellmann [2] developed a new field of cryptography called Public key cryptography, which made an enormous impact on directions and applications of cryptography. Public key cryptosystems allow all information including key to be distributed over an insecure communication channel without loss of confidentiality. The private keys remain with the original user and there is no need to transmit it, thereby increasing the security of the system. Public key cryptosystem also accommodate digital signatures so that sender of the message can be easily authenticated.

The map between plain text and cipher text for public key cryptosystems make use of the idea of a one-way function so that encryption using this function is computationally infeasible.

Some purposes where public key cryptography has been applied are

- Confidential message transmission.
- Authentication-which establishes that the message was sent by the person claimed and that it has not been tampered with.
- Non –repudiation – Which guards against people claiming not to have agreed to do something that they really agreed to.

- Key establishment-Where two people using the open airways want to agree upon a secret key for use in some symmetric key cryptosystem.
- Electronic cash mechanisms that ensure spender anonymity.
- Electronic Voting schemes that ensures that votes are correctly and confidentially trailed.

2. 1.2 Advantages

1. Security

The Primary advantage of Public key cryptography is increased security. The private keys used in this are not revealed to anyone.

2. Digital Signatures

Another major advantage is that they can provide authentication via secret key, it requires sharing of some secret information and sometimes requires trust of third party. A sender can then repudiate a previously signed message by claiming that the shared secret was compromised by one of the parties sharing the secret.

2.1.3. Different Public Key Cryptographic methods

In this section we will explain various cryptographic algorithms

2.1.3.1 RSA Cryptosystem

It is one of the widely known cryptosystem developed by R.Rivest, A.Shamir and L. Adleman[3,4]. It is used to provide secrecy and security. Its security based on integer factorization problem. RSA is widely used in electronic commerce protocols. Every cryptographic algorithm can be implemented in 3 steps

a. Key generation

b. Decryption

c. Encryption

a. Key generation

Since RSA is a public key cryptosystem it involves two keys –private key and a public key. Messages are encrypted using public key and decrypted using private key.

i. Generate two large prime numbers p and q.
ii. Compute n = p. q and $\varphi = (p-1).(q-1)$.
iii. Select a random integer e, $1<e<\varphi$.
iv. Determine d which satisfies d.e =1.
v. A's public key is (n, e) , A's private key is d.

b. Encryption

i. Obtain A's authentic public key (n, e).
ii. Represent the message as an integer m in the interval [0, n-1].
iii. Compute $c = m^e \pmod n$.
iv. Send the cipher text to A.

c. Decryption

To recover the plain text we should do the following

Use the private key d to recover the message plain text i.e. $m = c^d \pmod n$. One of the major problems in RSA is integer factorization problem. Multiplying two large numbers is easy in forward direction, but finding the numbers (or factors) in inverse is quite difficult. This problem is called integer factorization problem

Here is an example of RSA encryption with small parameters [3].

Key generation: A chooses prime p=2371 and q=2557 and computes n = pq=606217 and φ = (p-1)(q-1) =6057720. A then chooses e=367453 and by using extended Euclidean algorithm he finds d=4953277. Such that ed = 1 (mod φ). So A's public key pair is (n=6062617, e= 367453) and private key (d=4953277).

Encryption: Consider that our message m = 5234681. B uses modular exponentiation to compute c = m^e (mod n) =5234681^{367453} (mod 6062617) =5640058 and c is sent to A.

Decryption: To decrypt c, computes c^d (mod n) = $5640058^{4953277}$ (mod 6062617) = 5234681.Thus after the process of decryption we got the original message we transmitted.

RSA system security is based on Integer factorization problem. Integer factorization problem is the problem of factorization of very large numbers. Some other security issues related to RSA include the following [3, 4]

1. Factoring attacks: Given (n, e) as public information, one attack is to factorize n and thereby computing φ and d.
2. Small encryption Exponent e: If e is very small m can be recovered for c = m^e (mod n) via e^{th} root of c.
3. Forward search attack: If message space is small or predictable an adversary can decode the entire possible message set until it gets c.
4. Small decryption Exponent d : If d is small, we can easily compute d from publicly known e and n by using certain algorithms.
5. Multiplicative properties: Suppose we have two plain text messages m1 and m2 then, there are c1 and c2 such that (m1, m2) ^{e} =$m1^e$. $m2^e$ = c1. c2 (mod n). This is referred to as homomorphism properties of RSA [3], which lead to adaptive chosen text attack on RSA.

6. Common modulus attacks: Sometimes there may be a central trusted authority uses a single RSA modulus n and then in turn distribute a distinct encryption/decryption pairing (e_i,d_i) can lead to factorization of n. The factorization of n would lead to the discovery of all other key pairings that were generated by the original trusted authority.

2.1.3. 2 Rabin Public Key Encryption

The Rabin public key encryption scheme was the first example of provable secure encryption scheme[3].The problem faced by a passive adversary of recovering plain text from some given cipher text is computationally equivalent to factoring of integers

a. Key generation

i. Generate two large random prime p and q whose size is roughly same.
ii. compute n = p. q.
iii. A's public key is n, A's private key is (p ,q).

b. Encryption

i. Obtain A's authentic public key n.
ii. Represent the message as integer in the interval [0, n-1].
iii. Compute m^2 (mod n).
iv. Send the cipher text C to A.

c. Decryption

To recover the plain text we should do the following.

i. Using Extended Euclidean algorithm find 4 square roots m1, m2, m3, m4 of C modulo n^2.
ii. The message sent was m1, m2, m3 or m4.

Here is an example of Rabin public key encryption with small parameters [3]. For the purpose of key generation Entity A chooses two primes p= 277 and q=331 and n= p. q = 91687. A's private key is (p=277, q=3310) and public key is n (=91687).

Encryption: Convert message to be sent into bits. Consider our message is of 10 bits, we replicate last 6 bits of message if m^1 = 1001111001, then m=1001111001111001 i.e. now m=40569. B computes $c=m^2 \pmod n$ =62111 and sends it to A.

Decryption: To decrypt we find $4\sqrt{c} \pmod n$ then m1=40569, m2=22033, m3=40569, m4=51118 . In binary

m1=1000100000010110, m2=1010111000010001

m3=1001111001111001, m4=1100011110101110

Here m3 has redundancy. A decrypt to m and recovers the original message m3=100111001.

Security of Rabin public key encryption [3, 4] can be described as follows

1. The task faced by a passive adversary is to recover plain text m from the corresponding cipher text C. The problem of computing n and computing square roots modulo n is computationally difficult.
2. Chosen cipher text attack : The adversary select a random integer $m \in Z_n^*$ and computes $C = m^2 \pmod n$, The adversary then presents C to A's decryption machine which decrypt and return the plain text with a probability ½, $y \neq \pm m^2 \pmod n$, in which case gcd (m-y ,n) is one of the prime factors of n . If $y \equiv \pm m \pmod n$, then the attack is repeated with a new m^3.
3. It is also susceptible to the attack similar to those on RSA.

A drawback on this system is that receiver is faced with the task of selecting the plain text from among four possibilities. This problem is overcome by adding pre specified redundancy to the original plaintext prior to the encryption.

2.1.3 .3 The Diffie-Hellman Public Key exchange system

Diffie and Hellman [2] were the first to propose a solution to the key distribution problem and digital signature problems in 1976 They used what is known as trapdoor one-way function .A trapdoor one-way function is a special function which is easy to compute, However, given this type of mapping it is very difficult to find an inverse without another trap-door function.

a. Key Generation

i. A and B publicly select a finite group G with an element $\alpha \varepsilon G$.

ii. A generates a random integer a and computes $\alpha^a \varepsilon G$, public key is α^a. The key is exchanged by sending α^a to B.

iii. On the other hand B generates a random integer b such that $\alpha^b \varepsilon G$, and transmit α^b to A over the same channel. (Public key is exchanged).

b. Encryption

Let M be the message to be transmitted and M is represented as an integer. A computes $C = M. (\alpha^b)^a$ and C is sent to B.

c. Decryption

M is recovered by C. $((\alpha^a)^b)^{-1}$.

2.1.3 4 The Elgammal encryption

In 1985 Elgammal [6] proposed a public key cryptosystem based around the Diffie – Hellmann key exchange scheme. This scheme is based on discrete exponentiation which exhibits the properties of a trapdoor one-way function. This system is based on using the multiplicative group of a finite order Zp.

a. Key Generation

i. Generate a large random prime p and a generator α, a multiplicative group Zp^* of the integers modulo p.

ii. Select a random integer α, $1 \leq \alpha \leq p-2$ and compute $\alpha^a \bmod p$.

iii. A's public key is (p, α, α^a); A's private key is a.

b. Encryption

i. Obtain A's authentic public key (p, α, α^a).

ii. Represent the message as an integer m in the range {0, 1.....,p-1}.

iii. Select a random integer k, $1 \leq k \leq p-2$.

iv. Compute $\gamma = \alpha^k \pmod p$ and $\delta = m.(\alpha^a)^k$.

v. Send cipher text $C = (\gamma, \delta)$ to A.

c. Decryption

i. Use private key to compute γ^{p-1-a} .

ii. Recover the plain text m by computing $(\gamma^{-a}).\ \delta \pmod p$.

e.g. [11] Let A selects the prime p = 2357 and a generator α =2 of Z^*_{2357}. A chooses the primitive key a=1751 and computes $\alpha^a \pmod p = 2^{1751} \pmod{2357} = 1185$.

So A's public key is (p=2357, α =2, α^a = 1185) . Let our message m = 2035, B select a random integer K = 1520 and computes $\gamma = 2^{1520}$ (mod 2357) = 1430 nd δ = 2035 . 1185^{1520} (mod 2357) = 697. B sends γ=1430 and δ = 697 to A. To decrypt A computes γ^{p-1-a}= 1430^{605} (mod 2357) = 872 and recovers m by computing m= 872.697 (mod 2357) = 2035. i.e the message we sent.

Main advantage is that all entities must choose same prime p and generator α, in which p and α should not be chosen as a part of the public keys. This result in public keys of smaller sizes .An Additional advantage of having a fixed base α is that exponentiation can be expedited via pre computations.

Security of Elgammal encryption [3, 6] can be as follows.

1. It is based on discrete logarithm problem over a field Fp^*.

2. It is critical that different random integer K used to encrypt different message, suppose the same K is used to encrypt two messages m1 and m2 and the resulting cipher text pairs are (γ_1 , δ_1), (γ_2 , δ_2),. Then δ_1/δ_2 = m1/m2, m2 could be easily computed if m1 is known.

Definition 3.1: Discrete Logarithm Problem: - Given a prime p, a generator α of F_p^* , find the integer x , $0 \leq x \leq p-2$ such that $\alpha^x \equiv \beta \bmod p$.

Whenever we develop a system, the parameters should be chosen in such a way that the solution is infeasible.

2.1. 4 Elliptic Curve Cryptography

In 1985, Victor Miller [7] and N.Kobliz [8] independently, proposed a public key cryptosystem analogue of the Elgammal scheme, in which group Zp^* is replaced by a group of points on the elliptic curve defined over a finite field. The main attractions of elliptic curve cryptography over competing technologies such as RSA and DSA is that various algorithms are known for

solving the underlying hard mathematical problems in Elliptic Curve Cryptography .Elliptic curve discrete logarithm problem takes fully exponential time. On the other hand, the best algorithm known for solving the underlying hard mathematical problem in RSA and DSA (Integer Factorization problem and DLP problem) take sub-exponential time. This means that significant parameters used in ECC is small compared to RSA and DSA but with equivalent levels of security.

The lack of sub exponential attack on ECC offers potential reductions in processing power, storage space, band width and electrical power. These advantages are especially important in applications on constructed devices such as smart card, pagers, cellular phones etc.

The performance of ECC depends mainly on the efficiency of finite field computations and fast algorithm for elliptic scalar multiplication. Although numerous known algorithms are available for the elliptic curve arithmetic operations, the performance of ECC can be speeded up by selecting specific underlying finite field and/ or elliptic curve. ECC [9 10] is used in many areas. Construction of elliptic curve cryptosystems requires following steps.

1. Select an underlying curve.
2. Select a representation for the elements in F_q.
3. Implementation of arithmetic over F_q.
4. Select an appropriate elliptic curve $E(F_q)$.

The elliptic curve operations are implemented on $E(F_q)$. From these things we can see that elliptic curve system is dependent on two things. First one is finite field and its operations and the other, the elliptic curve and its operations.

Advantages

- Greater flexibility in choosing cryptographic system.
- Requires shorter keys.
- Greater speed and requires less storage space. Because of this ECC can be used in smart cards, Cellular phones, pages etc.
- Reduced band width and much more efficiency.
- Known theoretical attacks are less effective while using ECC.

Disadvantages

- Patented, uncertainty regarding their implementation.
- Algorithms are more complex, so quite difficult to implement.
- ECC is mathematically more subtle than RSA or DSA. That means difficult to explain or justify to the client.
- Hyper elliptic cryptosystem offer much smaller key size.

Main uses of ECC include key exchange, digital signature, authentication, message transmission etc.

2.1.4.1 Elliptic Curve Discrete Logarithm problem

The security of crypto system is dependent on hardness of Elliptic Curve discrete logarithm problem.

Definition 2.2[5]: The Elliptic Curve discrete logarithm problem (ECDLP) is, given an elliptic curve E defined over a finite field Fq, a point $P \varepsilon E(Fq)$ of order n, and a point $Q \varepsilon < P >$, find the integer $l \varepsilon [0, n-1]$ such that $Q = l.P$. The integer l is called the discrete logarithm of Q to the base P, denoted as $l = \log_P Q$.

The elliptic curve parameters for the cryptographic scheme should be carefully chosen in order to resist all known attacks on the ECDLP. The most

naïve algorithm for solving the ECDLP is exhaustive search where by one computes the sequence of points P, 2.P, 3.P....until Q is encountered. The running time is approximately n steps in the worst case and n/2 steps on average. Therefore, exhaustive search can be circumvented by selecting elliptic curve parameters with n sufficiently large to represent an infeasible amount of computation.

2.1.4.2 Diffie – Hellmann Key Exchange for elliptic curves

Here we will see how Diffie – Hellmann Key exchange is used along with elliptic curves. This will enable the people involved in communication or key exchange say for example A and B to securely construct a key for use in symmetric key encryption such as DES or AES.

Procedure is as follows.

i. A and B agree on an elliptic curve E over a finite field F_q , So discrete logarithm problem is hard in $E(F_q)$. They also agree on a base point $P \varepsilon E (F_q)$ such that the subgroup generated by P has a large order (Usually prime).
ii. A chooses a secret integer α, and compute $P_a = \alpha. P$, it then sends P_a to B.
iii. B chooses secret integer β, and then computes $P_b = \beta.P$ and sends P_b to A.
iv. A computes $\alpha P_b = \alpha .\beta.P$. The B computes $\beta P_a = \alpha. P_b$. A and B agree on a method to exchange key.

We will consider an example to illustrate this algorithm. Let E: = $y^2 = x^3 + 4$ defined over a field F_{121} and P(2,2) be an element of F_{121} . Both of these are agreed publicly by A and B. A then chooses a secret integer 113 and calculates $P_a = \alpha.P = (115,48)$. A sent P_a to B. Bob chooses a secret integer $\beta = 98$ and computer $P_b = \beta P = (130,203)$. B sends this P_b to A. A computes $\alpha P_b = (161,169)$ and bob computes $\beta.P_a = (161,169)$. There by A and B have

securely generated the points. So that, encryption and decryption process can be performed successfully.

2.1.4.3 El gammal crypto system for elliptic curves [7,8]

Here we will see how El gammal cryptosystem is used in cryptosystem using elliptic curves. Let F_q be a finite field and let B be a base point. The steps involved in this process include

a. Key Generation

i. Let a be random integer of Alice and b be random integer of Bob.
ii. Compute p = b.B. Make bB public. So public key is b.B.

b. Encryption

i. Let M be message to be transmitted. Convert M to a point on the elliptic curve. Let it be P_m.
ii. Alice computes cipher text as follows (a.B , P_m + a.p).
iii. This cipher text is sent to Bob.

c. Decryption

Bob receives cipher text and produces the message as follows.

Compute P_m + a. p – a. b. B, Where b is secret key of bob and p = b. B. Then Convert P_m to message M.

2.1.4.4 Massey-Omura Elliptic Curve Cryptosystems

This is another cryptosystem [7] which make use of fundamentals of elliptic curve and make use of advantages of elliptic curve cryptosystems. Now we describe the process involved in this cryptosystems

a. Key generation

Let F_q be a finite field and E be an elliptic curve

i. Let N be a publicly large prime number Alice choose a secret key c, such that $0 < c < N$ with gcd (c, N) = 1.
ii. Let Bob choose a secret key d with gcd (d, N) = 1.

b. Encryption and Decryption

i. Let M be the message to be transmitted. Convert message M to a point P_m by message embedding.
ii. Alice sends c. P_m to bob .
iii. Bob responds it with d.(c. P_m).
iv. Alice sends back $c^{-1}(d .c. P_m)$ such that $c^{-1}.c = 1$.
v. To decrypt Bob recovers message with $d^{-1}(d .P_m)$ by reversing the embedding P_m

2.1.4.5 Digital Signatures in ECC

Digital signature of a message is a number dependent on some secret known only to the signer. Signature must be verifiable. Digital signature produce more data integrity and non – repudiation to message. We can implement digital signature schemes in elliptic curve cryptography also. The Elliptic curve Digital signature Algorithm [5] is as follows. The algorithm can be described in three steps.

a. Key generation

Let F_q be a finite field and P be base point on elliptic curve. Generate a random integer d whose values are in between 1 and q.

b. Signature generation

i. Select a random integer k ε [1,n-1].

ii. Compute P = (x_1, y_1) and convert x_1 to integer x_1'.

iii. Compute r = x_1' .(mod n). If r = 0 go to step 1.

iv. Compute e = H (m).

v. Compute s = k^{-1} . (e + d. r) (mod n). If s= 0 go to 1.

vi. Signature is (r, s).

c. Signature Verification

i. Verify that r and s are integers between [1, n-1] if verification fails, reject.

ii. Compute e = H (m).

iii. Compute w = s^{-1} (mod n).

iv. Compute u_1 = e. w (mod n) and Compute u_2 = r.w (mod n).

v. Compute X = $u_1P + u_2 Q$.

vi. If (X = ∞) Return(" reject signature").

vii. Convert X to x_1' and to x_1. Compute u = x_1 (mod n).

viii. if u = r then accept or else reject.

These are the various cryptographic algorithms available for elliptic curve cryptography.

2.1.4.6 Security Level and Comparison of ECC with other cryptosystems

Security of a cryptosystem [9-11] very important because whenever we develop it should not be prone to attacks. Here we discuss certain factors that focus on security of the elliptic curve system. Since we are using elliptic curve, various factors in curve itself affect security of the system.

1. Selection of field

An elliptic curve is constructed on finite field. We can choose elliptic curve either on prime field F_q or F_2^m. F_q is prime finite field and F_2^m is binary extended finite field. Elliptic curve discrete logarithm problem is difficult to solve F_2^m than in F_q.

2. Representation of elements in F_q

Once the field F is selected, there are many ways of representing elements in the field. They include optimal normal basis representation and a polynomial basis representation. Since elements in one representation can be efficiently converted into elements in other representation by choice of basis matrix, then intractability of ECDLP is not effected by choice of representation.

3. Selection of elliptic curve over F_q

There are various ways for selecting elliptic curve. They include random selection method, Koblitz curve selection method and Complex multiplication method. Whatever method we choose it should satisfy certain constrains that is number of points on the curve should be divisible by a sufficiently large prime and also should satisfy the security level of an elliptic curve which is fixed and is $160 \leq L \leq \lfloor \log_2 q \rfloor$. This thesis uses Kobliz curve selection method for generation of secure curves. The method is given in Appendix B.IV.

The above levels are fixed in order to avoid various attacks on curves. Attacks include Pollard- Rho attack , Pohig-Hellman attack, index-calculus attack etc. If we are generating a curve according to the above mentioned parameters we can overcome the attacks and a stable system is generated and this will be more stable than other public cryptosystems.

Security Level of ECC compared with RSA and DSA [9, 10] can be represented by a graph. The graph representing security level of elliptic curve cryptosystems compared with RSA and DSA.

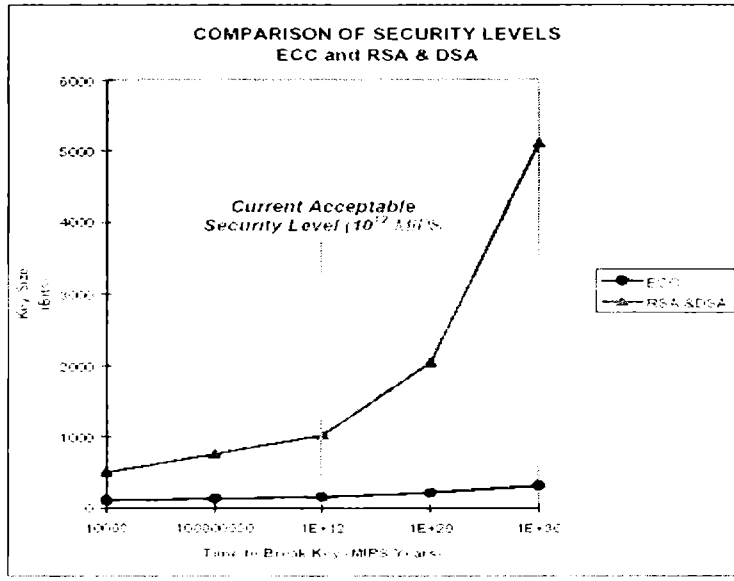


Fig 2.2 Security level of ECC and RSA

Elliptic curve cryptography has gained attention in recent years due to the ability to provide equivalent security as RSA at much smaller key sizes and at faster rates. Because of this ECC has been considered for applications such as smart card encryption due to less storage requirements and its computational efficiency. Table below shows a comparative study of key size in bits for equivalent levels of ECC with other cryptosystems. [9]

Symmetric	ECC	DH/DSA/RSA
80	163	1024
128	283	3072
192	409	7680
256	511	15370

Table 2.1 Key size comparison

The table 2.1 is a comparative study of key sizes of various cryptographic systems. Here we have taken three different types of cryptography which include Symmetric, Public and Elliptic Curve cryptography. In these Cryptosystems, public key cryptography is widely used now a days. When we compare them we can see that ECC key size is very less compared to public key cryptosystems. Attacks are very common in various cryptosystems. The table below shows attacks on keys of various sizes on RSA and ECC [10]

Time to break in MIPS years	RSA/DSA Key Size	ECC Key Size	RSA/DSA Key Size ratio
10^4	512	106	5:1
10^8	768	132	6:1
10^{11}	1024	160	7:1
10^{20}	2048	210	10:1
10^{78}	21000	600	35:1

Table 2.2 Comparison ECC with RSA and DSA

The Table 2.2 shows attacks on ECC key compared to RSA/DSA. In addition to speed, elliptic curve resists breaking by current number sieve method and index calculus method. We can conclude that elliptic curve cryptography will dominate cryptosystems in near future.

2.2 Public key Cryptosystems based on Codes

In the above chapters we have seen various public key cryptographic methods. Here a discussion on the cryptography based on codes is done. Codes here come from Coding theory which involves the error detection and correction. So here we are combining security with information reliability.

2.2.I Mc -Eliece cryptosystems based on linear codes

In 1978 Mc-Eliece [12] introduced a public key cryptosystem based on binary linear code. He suggested his scheme based on generator matrix of a [1024,524,101] Goppa code. The security of the system is based on the NP-completeness of the decoding problem for general linear codes. This scheme requires very large block length to allow introduction of large number of errors so as to force a high work factor. The computational overhead for the process of encryption and decryption is very high. The algorithm can be described as follows.

a. Key Generation

Each user A picks a K x N binary generator matrix G_A of a t-error correcting binary linear code and publishes the matrix as $G_A' = S\ G_A P$. G_A' is the public key. S here is a randomly generated K x K non-singular binary matrix and P is an N x N permutation matrix. G_A, P, S are private keys.

b. Encryption

Suppose M is the message to be transmitted, then M is converted as a binary vector of length K. Then process of encryption is done as follows.

$M' = M.\ G_A' + Z$, Where Z is a random error vector of length n that introduces t errors which is used for correction and detections of errors.

c. Decryption

Decryption is done by using a fast decoding algorithm. When A receives M' ,he first correct errors in $Z.P^{-1}$ and then by using S^{-1} , retrieves M.

Cryptographic attacks for the cryptosystems described above are as follows. The following attacks have been proposed on the system [12, 13, 14]. There are three possible attacks.

1. Structural attack [16]: This process involves getting S, G_A, and P from $G_A{}^1$. Once these parameters are obtained, an intruder can easily attack the system and get the message.

2. Recovering the message directly without the keys G or P or S: In this attack, method one tries to get k i.e. size of the message. Once the message size is obtained it tries to recover the message by solving K x K system of linear equations.

3. Decoding attacks: From the message received they will try to decode the message to get the original message by making use of available parameters. The complexity of decoding an arbitrary q-ary linear code with errors having arbitrary q array values is much higher than complexity of decoding a code with comparable parameters.

2.2.2. Niederreiter cryptosystems

This cryptosystem was proposed by Prof. H.Niederreiter in 1986 [13]. It is based on Reed-Solomon code discussed in section 1.2.2. The procedure for this method is as follows

a. Key generation

i. The parity check matrix H of a generalized Reed-Solomon code is used here instead of generator matrix.

ii. A non-singular scrambling matrix S of order n is used. This is used to scramble the parity check matrix i.e. to destroy any evident structure of the parity check matrix.

iii. S and H are private keys.

iv. Public key is $H^1 = S.H.$

b. Encryption

i. The plain text M is converted into a vector of size n.

ii. The cipher text c= mH^1.

c. Decryption

Once the cipher text is received, the user multiplies it with S^{-1} and will get m. H. This is the syndrome of plaintext m and by using fast decoding algorithm we will get plain text vector m which in turn can be converted into message M.

Attacks on Niederreiter Crypto System can be described as The Sidelnikov – Shestakov attack[15] . The Niederreiter system was broken by Sidelnikov – Shestakov. Everybody knows the public key H^1. But no one knows S and H separately. The breaking party tries to find trapdoors H_{tr} and S_{tr} such that $H_{cr} = SH = H_{tr}S_{tr}$ where $H_{tr} = [\ y_i\ \beta_j^i\ ]$ .The elements $\{y_i\}$ and $\{\beta_i\}$ may differ from the elements $\{x_i\}$ and $\{a_i\}$ of H_{cr} . Nevertheless, they allow decrypting any cipher text.

2.2.3 Analysis of the System

This section deals with comparison of Mc-Eliece with Niederretier cryptosystem based on linear code on key size and work function.

PKC	Parameters	Size of public key	Work function
Mc Eliece	Binary, n=1024,k=524,t=50	Large:5 x 10^5	$>2^{59}$
Niederreiter	n=128,d=64	32000	$O(n^3)$

Table 2.3 Comparison -Mc-Eliece with Niederretier

From Table 2.3 we can see size of keys of these two system is very high compared to other public key cryptosystems.. The Mc-Eliece Public key

Cryptosystem [16] seems to be secure even without any modification. Reason is that Goppa codes are sub codes over a finite field F_2 and there exist too many Generalized Reed Solomon code(GRS) that containing them as sub codes . Thus there is no evident way of finding a Goppa polynomial or GRS code from a scrambled matrix.

Niederreiter on the other hand is defined over a large alphabet. The weakness of such kind of cryptosystems is due to the very regular structure of the generator or parity check matrices, even if scrambled. The hiding of the public key by means of adding carefully chosen matrices prevents known attacks and provides security to this public key cryptosystems. Later various researchers studied about the system and various modifications were made on it. Mc-Eliece system was developed on Algebraic Geometric code also. In this thesis we are concerned with Algebraic Geometric code over Elliptic curves. The table below shows the parameters used by Mc-Eliece and the Algebraic Geometric code using elliptic curves.

Result	**Mc-Eliece code**	**Elliptic code**	**Elliptic code**	**Elliptic code**
(n ,k, d)	[1024,525,101]	[171,81,90]	[313,101,112]	[995,451,544]
Field	GF(2)-binary	GF(157)	GF(303)	GF(997)
No.of correctable errors	50	45	56	271
Message size	524	81	101	451

Table 2.4 Comparison of Mc-Eliece code and elliptic codes

The Table 2.4 is a comparison which is done with the help of MATLAB program for a standard Mc-Eliece code to the elliptic code. Result shows that elliptic codes are better than standard Mc-Eliece code in decoding point of view.

2.3 References

[1] Shueling Chang, Hans Eberle, Vipul Gupta, "Elliptic Curve Cryptography-How it works", Sun Microsystems.

[2] W.Diffie, M.E Hellmann, "New directions in cryptography", IEEE transactions for Information Theory.Vol-22, pp.644-654, 1976.

[3] Menezes A.J, Van Oorschot ,P.Vanstome, "Handbook of Applied Cryptography", CRC press,1997.

[4] B.Schneir , "Applied Cryptography 1996", Second edition, Wiley.

[5] Hankerson, Menzes, Vanstone, "Guide to elliptic curve cryptography, Springer", CRC press, 2004.

[6] Elgamal T, "A public key cryptosystem and a signature scheme based on discrete logarithm problem", IEEE Transactions on Information Theory Vol-31, no.4, 1985.

[7]. V. Miller, "Uses of elliptic curve in cryptology", Proceedings of crypto'85 LNCS 218, pp 417-426, New York: Springer-Verlag 1986.

[8]. N.Koblitz, "Elliptic curve crypto systems", Mathematics of computation, 48, pp.203-209, 1987.

[9] "Remarks on Security of ECC", a Certicom white paper published sep-1997.

[10] "ECRYPT Yearly Report on Algorithms and Key sizes (2004)", Document D.SPA.10, March 2005.

[11] Certicom Research, "SEC 2: Recommended Elliptic Curve Domain Parameters", Standards for Efficient Cryptography, Version 1.0, Sep. 2000.

[12] V.I.Korzhik and A.L Turkin , "Cryptanalysis of Mc-Eliece 's public key cryptoystems", Eurocrypt'91,Lecture notes in Computer Science, 547 ,pp 68-70, 1991.

[13]. H.Niederreiter "Knapsack type cryptosystems and Algebraic Coding Theory", Probl. Control and Information theory ,Vol.15, pp 19-31, 1986.

[14] R. J. McEliece, "A Public-key cryptosystem based on Algebraic Coding Theory", *DSN Progress Report*, Jet Propulsion Laboratory, Pasadena, CA (Jan./Feb. 1978) pp. 114–116.

[15]. V.M Sidelnikov, S.O Shestakov, "On the insecurity of cryptosystem based on generalized Reed-Solomon codes", Discrete Math; Vol.no.4 pp 439-444, 1992.

[16]. C. M. Adams and H. Meijer, "Security-related comments regarding McEliece public-key cryptosystem" Advances in Cryptology-CRYPTO' 87, Springer-Verlag, New York pp. 224–228 , 1987.

.....

Chapter 3

Secret Sharing

Contents

3.1 Secret Sharing

Secret sharing scheme [1] involves the construction of a secret, production of shares and distributing shares among various users. The secret may be recovered only by certain predetermined groups. Secret sharing protects secrecy and integrity of information (secret s). S will be referred to as the secret and $I_1, I_2, \ldots I_n$ will be referred as shares of S. The set from where the secrets are chosen will be denoted by S or by S_0 and the set of the shares assigned to the i^{th} user will be denoted by S_i for all $1 \leq i \leq n$.

A secret sharing scheme is coordinated by a dealer who has to be a mutually trusted party. But there are secret sharing scheme which can be configured without presence of dealer. The dealer receives this value, derives the corresponding shares and distributes them to the users. Thus there are two phases in secret sharing

1. Share – The dealer D associates with any player P, a secret a_i and broadcast this information. Construction of a_i depends upon the various secret sharing methods.

2. Reconstruct – The reconstruction of the secret can be made by the participants after they pool together their shares or by a special party, called combiner after receiving the shares from the users of an authorized groups.

3.1.1 Access structure

The access structure of a secret sharing scheme is the set of all groups which are designated to reconstruct the secret. The elements of access structure will be referred to as the authorized groups and rest is called unauthorized groups. The qualified groups are authorized to reconstruct the information about the secret.

As Ito, Saito and Nishizeki [2] have remarked that any access structure must satisfy the condition

$(\forall\ B \in P(\{1,2,\ldots,n\}))\ (\exists A \in A)(A \subseteq B) \Rightarrow B \in A)$ .This means that if a group can recover the secret a larger group can also recover the secret. This is called Monotone access structure [3].A monotone authorized access structure is

$$A_{min} = \{ A \in A \mid \forall B \in A \setminus \{A\}(\neg (B \subseteq A))\}$$

The rank of access structure is defined as maximum (minimum) number of the participants in a minimal authorized group.

3.1.2 Models of secret sharing

A secret sharing scheme is a method of splitting a secret into shares such that secret can be determined only by the authorized sub groups. Depending upon the quantity of information [4, 5] leaked to an unauthorized group secret sharing can be classified as

- Perfect secret sharing schemes: The shares of any unauthorized group give no information about the secret.
- Computational-secure secret sharing schemes: Some information about the secret is leaked to the unauthorized groups, but the problem of finding a secret is intractable.

There are various models for secret sharing. They are

a .Brickell-Davenport Model

Brickell and Davenport have proposed an elegant model for secret sharing [4]. Here secret sharing scheme is represented as a matrix M with some special properties. The Matrix M has n+ 1 column, the first one corresponding to the dealer and the rest corresponding to the users. For i ε { 0,1...n} and Let S(i) = { $M_{r,i}$ | r is a row in M}, Then S = S(0) and S_i = S(i) , for all $1 \le i \le n$. The dealer chooses an element s in S and a row r of M such that $M_{r,0}$ = S. The matrix M is public and r is private.

b. Brickell-Stinson Model

It is another model in which secret sharing scheme can be represented as a special set f of distribution rules. A distribution rule is a function $f:\{0,1,.,.n\} \rightarrow S \cup_i'' S_i=1$ such that f(0) ε S and f(i) ε S_i for all $1 \le i \le n$. If S_0=S, then a secret sharing scheme can be viewed as a special subset of the product of the family (S_i | S_i ε {0, 1...n}. An element f ε F represents a possible distribution of shares to the users, where f (0) is the secret and f (i) is the share corresponding to the i^{th} user, for all $1 \le i \le n$.

c. Entropy based model

This model used concepts of entropy for measuring the quantity of uncertainty about the secret. It is based on the below specified definition.

Definition 3.1[6] Suppose we have n users labeled 1,….,n and consider a set of groups $A \subseteq P\{1,2,.,n\}$. A perfect A –secret sharing scheme is a collection of random variables (S, I_1,I_2……,I_3) such that

- (Correctness)-for any A ε A, H(S| {I_i| i ε A})=0;
- (Security)-For any A ε A , H (S|{I_i | i ε A}) = H(s).

In a non-perfect secret sharing scheme, the second item is replaced by

For any A ε A , H(S| I_i| i ε A)>0.

3.1.3 Different Secret Sharing Methods

So far we have seen the models of secret sharing. Now we will see different methods of secret sharing. Generally speaking there are nine methods of secret sharing methods available till date. They are

i. Threshold secret sharing
ii. Unanimous consent schemes
iii. Secret sharing for graph based structure
iv. Weighted threshold secret sharing scheme
v. Hierarchical secret sharing scheme
vi. Compartmented Secret sharing schemes
vii. General Secret sharing schemes
viii. Ramp Secret sharing schemes
ix. Construction based on decompositions

Next section we will see a brief discussion about each and every method specified above.

i. Threshold secret sharing scheme

The secret sharing scheme in which the number of participants in the reconstruction phase plays a prominent role in recovering the secret is called threshold secret sharing scheme. Most famous and successful scheme in this scheme is Shamir's scheme [1]. Shamir's scheme is based on polynomial interpolation .Given any k pairs$(x_1,y_1),\ldots..,(x_k, y_k)$ with $x_i \neq x_j$ for all $1 \leq i < j \leq k$, there is one and only one polynomial P(x) of degree k-1 such that $P(x_i)$ and $P(x_i) = y_i$, for all $1 \leq i \leq k$. A polynomial P of degree k-1 is chosen and secret S is the coefficient free portion of the polynomial. The shares $I_1, I_2 \ldots I_n$ are chosen as $I_i = P(x_i)$, for all $1 \leq i \leq n$. Having the shares, secrets can be nb reconstructed using Lang rage's interpolation formula as

$$S = \Sigma_{i\in\Lambda}\left(\ell_i \prod_{j\in\Lambda(j)}\right)\frac{x_j}{x_j - x_i}$$

Various secret sharing schemes based on threshold scheme concept are available. Secret sharing based on Chinese remainder theorem, based on information dispersal, based on special categories of integers etc. Detailed discussions on these are beyond the scope of this thesis.

ii. Unanimous consent schemes

The condition in which $A=A_{min} = \{1, 2,\ldots,n\}$ is referred to as Unanimous consent schemes of rank n. In this scheme apart from threshold secret sharing schemes knowledge of shares of all users is required in order to recover the secret. A Unanimous consent schemes is equivalent to (n, n) threshold secret sharing scheme. Karnin, Greene and Hellmann have proposed a very simple Unanimous consent schemes [7]

- The secret S is chosen as a random number from a set of real numbers.
- The dealer generated the shares I_i as random numbers from the set, for all $1 \leq i \leq n-1$ and $I_n = S - \sum_{i=1}^{n-1} I_i \pmod n$.

- The Secret can be reconstructed as $S = -\sum_{i=1}^{n-1} I_i \pmod n$.

iii. Secret sharing for graph based structure

An access structure in which all minimal access set has two elements can be referred to as graph based access structure. A result developed by Brickell and Davenport [4] can be specified as follows

Theorem 3.1: Let G be a connected graph. Then there is an ideal secret sharing scheme for the access structure specified by G if and only if G is a complete multipartite graph

The access structure is specified by graph $K_{n1,n2,\ldots,nl}$. Let $V_1,\ldots,V_l$ be the vertices of graph $K_{n1,n2,\ldots,nl}$. The dealer chooses the pair wise distinct element $x_1, x_2 \ldots x_l \in GF_q$. The shares corresponding to some secret $S \in GF_q$ will be defined as $I_i = x_j S + r$, for all $i \in V_j$ and $1 \le i \le l$, where r is an arbitrary fixed element from GF_q. Any two users u_1, u_2, $u_1 \in V_{j1}$, $u_2 \in V_{j2}$, $j_1 \neq j_2$ can obtain the secret S as $S = (I_{u1} - I_{u2})(x_{j1} - x_{j2})^{-1}$.

iv. Weighted threshold secret sharing scheme

In this method a positive weight is associated with each user and the secret can be reconstructed only if sum of the weights of the participants is greater than or equal to a fixed threshold [1]. The weighed threshold scheme can be explained by following definition

Definition 3.2[1] Let $n \ge 2$, $\omega = (\omega 1, \ldots, \omega_n)$ be a sequence of positive integers, and a positive integer w such that $2 \le w \le \sum_{i=1}^{n} \omega i$. The access structure A is defined as

$$A = \{A \in P\{1,2,\ldots,n\} | \sum_{i \in A} \omega 1 \ge w \}$$

is referred to as (ω, w, n) –weighted threshold access structure. The parameters $\omega 1 \ldots\ldots \omega 2$ is referred to as weights and w is referred to as threshold of the

scheme. A (k, n) –threshold secret sharing scheme is a (ω, w ,n) weighted secret sharing scheme with weight $\omega_1 = \omega_2 = \ldots\ldots \omega_n = 1$ and w = k.

v. Hierarchical secret sharing scheme

As the name indicates in this method secret sharing is done n different levels $L_1, L_2 \ldots\ldots L_n$. A level threshold k_j is specified to the j^{th} level, for all $1 \leq j \leq m$. The secret can be reconstructed if and only if there is a level such that number of participants from this level or higher level is greater than or equal to initialization level threshold. This structure can be defined as follows.

Definition 3.3: Let $L = \{L_1, L_2 \ldots\ldots L_m\}$ be a partition $\{1,2 \ldots n\}$ and let us consider a sequence level threshold $K=(k_1, k_2, \ldots\ldots k_m)$, where $1 \leq k_j \leq |L_j|$, for all $1 \leq j \leq m$ and $k_i < k_2 \ldots\ldots < k_m$. The *(L, K)* multilevel access structure is given by

$$A = \{A \in P(\{1,2,\ldots n\}) \mid \exists j = \overline{1,m}) (|A \cap \cup_{i=1}^{j} L_i| \geq K_j)\}$$

This scheme can be called as (L , K) –multilevel secret sharing scheme.

vi. Compartmented secret sharing

In this secret sharing method , the set of users are partitioned into compartment $\{C_1, C_2, \ldots\ldots C_m\}$.Here a threshold k_j is applied the j^{th} compartment for all $1 \leq j \leq m$. The secret can be recovered if and only if the number of participants from any compartments is greater than or equal to the corresponding compartment threshold, and total number of participants is greater than or equal to the global threshold.

The access structure of the compartment secret sharing is as follows

Definition 3.4[8]: Let $C = \{C_1, C_2, \ldots\ldots C_m\}$ be a partition of $\{1,2,3 \ldots n\}$ and let us consider a sequence of compartment thresholds $K=(k_1 \ldots k_m)$ where $1 \leq k_j \leq |C_j|$, for all $1 \leq j \leq m$. The *(C,K,*k)- compartment access structure is given by

$A=\{A \varepsilon P(\{1,2,\ldots,n\}) \mid (|A| \geq k) \wedge (\forall j = \overline{1,m})(|A \cap C_j| \geq k_j)\}$. In this case a A-secret sharing scheme is referred to as a (C, K, k) – compartmented secret sharing. A (k, n)- threshold secret sharing scheme is similar to compartment secret sharing with C = {1,2...n}(m=1) and $K=(k, k)$.

vii. Ramp secret sharing

In these methods shares of smaller size is used. It provides a semi access groups who can obtain some information about secret. Thus this scheme may provide a compromise between the level of security and size of shares. If $n \geq 2, 1 \leq k \leq n$ and $1 \leq l \leq k$, A (l, k, n) – threshold Ramp scheme[9] is a method of generating $(S,(I_1, \ldots,I_n))$ such that

- For any $A \varepsilon P(\{1,2\ldots.n\})$ such that $|A| \geq k$, the problem of finding the element S, given the set $\{I_i \mid i \in A\}$, is easy.
- For any $A \varepsilon P(\{1,2\ldots.n\})$ such that $k - l + 1 \leq |A| \leq k - 1$, some information about S, can be found having $\{I_i \mid i \varepsilon A\}$.
- For any $A \in P(\{1,2\ldots.n\})$ such that $|A| \leq k - 1$, some information about S, can be found having $\{I_i \mid i \in A\}$.

Blakely and Meadows [9] showed that Shamir's threshold secret sharing scheme can be transformed into a threshold Ramp scheme by choosing secret S as a vector $(P(x_1),\ldots\ldots(P(x_n))$ instead of P(0). Linear Ramp scheme [9] are ramp scheme in which amount of secret information obtained by a semi-access group grows linearly with respect to the size of group. That is a linear (l, k, n) – threshold ramp scheme is the collection of random variables $(S, I_{1\ldots}, I_n)$ such that

- For any $A \varepsilon P(\{1,2\ldots.3\})$ such that $|A| \geq k$, $H(S| i \varepsilon A)=0$.
- For any $A \varepsilon P(\{1,2\ldots.3\})$ such that $k - l + 1 \leq |A| \leq k-1$, $H(S| I_i | i \varepsilon A) = (k - |A|)/ l \; H(S)$.

- For any A ε P ({1,2....3}) such that |A| $\leq k$-1, H(S| I_i |{ i ε A}) = H(S).

viii. Constructions based on decompositions

In this scheme secret sharing scheme is used as a decomposition of larger schemes. Martin [10] developed a method based on distribution rules .Any method that uses matrix for the purpose of secret sharing can be used for decomposition schemes. First Colum of the matrix is used as secret S and remaining columns are used as shares. Access structure that is defined by Stinson [11] is as follows

Definition 3.5 [11] : Let A be an access structure and λ -decomposition of A is a sequence $A_1, A_2 \ldots A_m$ such that $A_j \subseteq A$, for all $1 \leq 1 \leq m$ and for any $A \varepsilon A_{min}$, there are $1 \leq i_1 < i_2 < \ldots < i \leq m$ such that $A \varepsilon A_l$, for all $1 \leq 1 \leq \lambda$.

Stinson has proven that if the closures of the access structures from the decomposition of a certain access structure can be realized, then that access structure can be also realized. If we consider access structures A and let $A_1, A_2 \ldots A_m$ be a λ - decomposition of A such that there exists F^1, a $cd(A_j)$ - secret sharing scheme, having the set of secret sharing schemes $S_{j,o}$ =GFq, for all $1 \leq j \leq m$. Suppose that there exists some vectors $v_1, v_2 \ldots v_m \varepsilon GFq^{\lambda}$ such that (*){v_j| A ε A_j } generates ε GFq^{λ} for all A ε A_{min}.

3.2 Secret Sharing based on Algebraic Geometric code

In 1981 M.J Mc-Eliece and D.J Sarwate [12] found that Shamir secret sharing scheme was closely related to the Reed-Solomon code. In this section we will discuss how secret sharing can be applied to algebraic geometric codes. Idea used by [12] is to encode the secret into a codeword ($D_1 \ldots D_n$). By using the concept of coding theory by if we know D_1 remaining D's can be found out and reconstruct the secret.

3.2.1. Massey secret sharing scheme

This scheme is also referred as linear secret sharing scheme since it is based on linear codes [12, 13, 14]. Let $C \subset F_q$ be a k-dimensional linear code with generator matrix G. Throughout this we have to assume that G has no zero column. Secret S is an element of F_q and their shares are distributed among n-1 entities and a dealer .In order to determine the shares, the dealer chooses $t \varepsilon C$, $t = (t_0 \text{------} t_{n-1})$ such that $t_0 = S$. He can choose such a t by first picking randomly a vector $u = (u_0, \text{------} u_{k-1}) \varepsilon Fq^k$ such that $s = ug_0$. Such a u can be chosen in q^{k-1} ways. Now t can be computed as $t = uG$. Shares are $\{t_1,.....t_{n-1}\}$ and G is shared. Only assumption made here is G can not have any zero column because if a column g_i is zero then t_i which is the share of the i^{th} participant will be zero. Hence this shareholder would not participate at all.

Since the columns of a generator matrix are linearly independent then the secret can be recovered by first solving the linear equation

$$g_0 = \sum_{j=1}^{m} x_j g_j$$

after finding x_j' s, the secret can be computed as

$$t_0 = ug_0 = \sum_{j=1}^{m} x_{jU} g_j$$

From now on we will assume that this is the only way to recover the secret for any set of shares.

3.2.2 Linear secret sharing in Algebraic Geometric Code on elliptic curves

Secret sharing on elliptic curve can be explained with the help of the theorem mentioned below.

Theorem 3.2 [16] Let E be an Elliptic curve over GF (q) with the group of GF (q) rational points E(GF(q). Then E (GF(q)) is isomorphic to $Z_{n1} \oplus Z_{n2}$. where n1 is a divisor of q-1 and n2.

Main thing in the secret sharing algorithm is the access structure of secret sharing algorithm. The access structures of elliptic secret sharing schemes [15] are based on some basic concepts of Algebraic-Geometric (AG) codes. Let X be a absolute irreducible, projective and smooth curve defined over GF(q) with genus 1 . D = $\{P_0, \ldots, P_n\}$ be a set of GF(q)-rational points of X and G be a rational divisor satisfying supp(G) $\cap$ D = φ.

Let L(G) = $\{ f : (f) + G \geq 0\}$ be the linear space (over GF(q)) of all rational functions with divisor not smaller than G, and (B) = $\{ \omega: (\omega) \geq 0 \}$ be the linear space of all differentials with divisor not smaller than B. Then the functional AG (algebraic-geometric) code $C_L(D;G) \varepsilon$ GF(q) and residual AG (algebraic-geometric) code $C_\Omega(D;G) \varepsilon$ GF(q)are defined as the evaluations of L(G) and $\Omega(G)$, respectively, at the points in the set D.

C_L (D; G) is a [n + 1, k = dim (L(G))-dim(L(G-D),d$\geq$ n + 1 - deg(G)] code and C_Ω (D; G) is an [n+1; k = dim((G-D))-dim((G)); d = deg(G)-2g +2] code

over GF(q). Then C_L(D;G) and C_Ω (D;G) are dual codes. Using C = C_L(D;G), secret sharing schemes based on AG codes were constructed in [7]. From the results in [7], it is known that in the case of elliptic secret sharing schemes, i.e., where X = E is an elliptic curve and the genus g = 1, every subset with at least n-deg(G)+2 elements is qualified and every subset with fewer than n - deg(G) elements is unqualified. In the following result, we determine explicitly which sets with n-deg (G) or n-deg(G)+1 elements are qualified. In general, any qualified set is said to be minimal if none of its proper subsets is also a qualified set. We also note that, when X = E is an elliptic curve over GF(q), the set of GF(q)-rational points on E, denoted by E(GF(q)), forms a finite abelian group with zero element O.

Theorem 3.3: Let E be an elliptic curve over GF(q). Let D = $\{P_0, P_1, \ldots, P_n\}$ be a subset of E(GF(q)) of n + 1 nonzero elements and let G = mO. Consider the elliptic secret sharing scheme obtained from E with the set of players P = $\{P_1, \ldots, P_n\}$. Let A = $\{P_{i1}, \ldots, P_{it}\}$ be a subset of P with t elements, and let B be the element in E(GF(q)) such that the group sum of B and $\{P_{i1}, \ldots P_{it}\}$ in E(GF(q)) is O. If A^c = P / A is a minimal qualified subset for the secret sharing scheme from C (D; G), then t ≤m. Furthermore,

1) When t = m, A^c is a minimal qualified subset if and only if B = O.
2) When t = m – 1, A^c is a minimal qualified subset if and only if B is not in D or B is in the set A.
3) Any subset of P of more than n -m+ 2 elements is qualified.

An example can be illustrated as follows. Consider an elliptic curve over a field F_9 and $y^2 = x^3 + x + 4$. Then E (F_9) is a cyclic group of order 10 with O the point at infinity and the points on curve are

$P_0(4,0), P_1(4,3), P_2(4,6), P_3(6,1), P_4(6,8), P_5(0,2), P_6(0,7), P_7(3,4), P_8(3,5)$

Here P_0 is the generator of E(F_9), then the points satisfy the condition $P_i = (i+1)P_0$ are D = $\{P_0, P_1, P_3, P_5, P_7\}$. Then access structure contains $\{P_1, P_3, P_5, P_7\}$ and

a. All subsets of P with 3 elements in P
b. The subset of 2 elements $\{(P_1, P_7), (P_1, P_3), (P_1, P_5), (P_3, P_5), (P_3, P_5), (P_5, P_1)\}$

By using these access structures, secret sharing method can be implemented for Algebraic geometric code using Elliptic Curves.

3.3 Refrences

[1] A. Shamir, "How to Share a Secret", Communications of the ACM, 22(11), pp.612-613, 1979.

[2] M.Itom, A.Saito and T.Nishizeki, "Secret sharing scheme realizing general structure", Proceedings of the IEEE global telecommunication conference, Globecom'87,pp 99-102, IEEE press 1987.

[3] J.Benaloh and J.Leichter, "Generalized Secret Sharing and Monotone functions", Advances in Cryptology-Crypto '88, Lecture Notes in Computer Science, pp-27-35, Springer Verlag, 1989.

[4] E.F Brickell and D.M Davenport, "On the Classification of Ideal Secret Sharing Scheme", Journal of Cryptology,Vol. 4(2), pp.123-134,1991.

[5] E.F Brickell and D.R Stinson," Some Improved bounds on the information rate of perfect Secret Sharing Schemes", Journal of Cryptology 5[3], pp.153-166,1991

[6] B.Blakley, G.R Blakley, A.H Chan and J.L Massey, "Threshold Secret Sharing Scheme with disenrollment" , Advances in Cryptology – Crypto' 92, Volume 740, Lecture Notes in Computer Science, pp.540-548, Springer Verlag,1993.

[7] E.D Karnin, J.W Griene and M.E .Hellman, "On Secret Sharing Systems", IEEE transactions on Information Theory,pp. 35-41, 1983.

[8] G.J .Simmons, "How to (really) Share a Secret", Advances in Cryptology – Crypto'88, Vol.403, Lecture Notes in Computer Science, pp 390-448, Springer – Verlag, 1990

[9] G.R Blakley and C.M, "Security of Ramp Schemes", Advances in Cryptology – Crypto'84, Vol.196, Lecture Notes in Computer Science, pp. 242-268, Springer – Verlag, 1985.

[10] K.M.Martin, "New Secret Sharing Scheme from Old", Journal of Combinatorial Mathematics and Combinatorial Computing, 14, pp 65-77, 1993.

[11] D.R Stinson, "Decomposition Constructions for secret Sharing Scheme", IEEE transactions of Information Theory, 40(1), pp.118-125,1994.

[12] R.J Mc-Eliece , "On Sharing Secrets and Reed – Solomon Codes", Communications of ACM, Vol.24, No.9, Sep 1981.

[13] Hao Chen, "Linear secret sharing from Algebraic Geometric codes", arxiv.cs/0603008v4[cs.cr],March 2006.

[14] Hao Chen, "MDS ideal secret sharing scheme from Algebraic Geometric code on Elliptic Curves", arxiv.cs/0608055v2[cs.cr] ,Sep 2006.

[15] Jun Xu, Xiaomin Zha, "Secret Sharing Schemes with general access structure based on MSP's", Journal of communications, Vol.2, Jan 2007.

[16] Hao Chen, San Ling and Chaoping Xing, "Access Structures of Elliptic Secret Sharing Schemes", IEEE Transactions on Information Theory, Vol.54, NO 2, February 2008.

.....

Chapter 4

Methodology

Contents

4.1 Theoretical Aspects of the problem

4.1.1 Introduction

Cryptography and Coding theory are specified as two hands in transmission of information. The Sender sends the information via communication channel. Communication channel is not error free. Lots of errors will be there due to noise and other disturbances in the channel. Information will be grabbled and combined with errors. While we transmit information, we will be sending secured passwords, secured banking information etc. There is a possibility that a third party reads and seizes these information and uses it. An intruder may change it and sent it through the communication channel. It means secrecy of our information is compromised. In order to overcome this problem two separate branches of science had been developed: - Coding theory and Cryptography. Coding theory involves sending information in coded form and decoding at receiving end, so that error correction and detection can be done. Cryptography deals with secrecy of information. Information is encrypted and sent and at the receiving end information is decrypted. By sending like this secrecy of information is preserved. A third party who is trying to read the message can see only the

encrypted information so that it will be very difficult to access the information transmitted. Here in this thesis we are combining Coding theory and Cryptography, thereby ensuring secure error free information being transmitted and received.

4.1.2 Concepts

Main concepts used in the system here includes

a. Finite field

Finite field [1, 2] is a field with finite number of elements. Finite field can be represented in prime field, binary field and binary extension field.

- Prime Field: The field is represented in the form F_P and it contains integers of the form {0, 1, 2....p-1} and contains p number of elements.
- Binary Field: The field is represented in form F_2^m and it contains binary elements The set of elements are of the form { $\alpha_0, \alpha_1 \ldots \alpha_{m-1}$ }such that $a_i \varepsilon \sum_{i=}^{m-1}{}_0\ a_i\ \alpha_i$, where $a_i \varepsilon \{0,1\}$.The set $\{\alpha_0 \ldots \alpha_{m-1}\}$ is called basis of F_2^m over F_2 .
- Extension field: The field of the form F_2^p. Elements are within binary field containing p elements.

Representation of elements in binary and extension field can be in two ways.

1. A normal basis representation of F_2^m over F_2 is a basis of the form $\{\beta, \beta^2, \ldots\ldots, \beta^{2m-1}\}$ where $\beta \varepsilon F_2^m$. So every element a is usually denoted by the string (a_0 , $a_1 \ldots\ldots a_{m-1}$) of length m . A normal basis representation of F_2^m has the computational advantage that squaring an element is a simple cyclic shift of the vector representation.

2. Polynomial basis Representation:Let $f(x) = x^m + \sum_{i=0}^{m} f_i x^i$ where $f_i \varepsilon \{0,1\}$ for i = 1 to m-1 be an irreducible polynomial. For each polynomial, there exists a polynomial basis representation. In such a representation each element F_2^m corresponds to binary polynomial of degree less than m.

In the proposed algorithm we have used finite field over prime field.

b.Curves

Algebraic varieties[5]: Let K be a field and $f_1,\ldots f_s$ be polynomials in $K[x_1,x_2,\ldots x_n]$, Then we have a set $V(f_1,\ldots f_s) = \{(a_1,\ldots a_n) \varepsilon K^n : f_i(a_1,\ldots a_n) = 0$ for all $1 \leq i \leq s\}$ we call $V(f_1,\ldots f_s)$ the affine variety defined by $\{f_1,\ldots f_s\}$. Thus we can use the term affine variety $V(f_1,\ldots f_s) \subset K^n$ is the set of solutions of the equations $f_1(x_1,\ldots x_n) = f_s(x_1\ldots x_n) = 0$. This variety will have dimension and variety of dimension one is called curve and variety of dimension 2 is called surface. A non-Singular Curve [5] can be defined as follows ,Let F be a curve, $P = (a, b) \varepsilon F$, P is called a simple point of F if either derivative $F_x(P) \neq 0$ or $F_y(P) \neq 0$ .In this case the line $F_x(P)(x-a) + F_y(P)(Y-b)$ is called the tangent line to F at P. A point which is not simple is called singular. A curve with many simple points is called non singular curve. Every non-singular curve over C can be realized as a surface R^3. For example an elliptic curve has an equation of the form $y^2 = f(x)$ where f(x) is a cubic polynomial in x with no repeated groups and can be thought as a torus[5, 6] in surface R^3 . In general, every non-singular curve can be realized as a torus with some number of holes. The number of holes in a curve is called genus of the curve. Thus the genus of a curve is a non-integer indicating the twistedness of a curve. The higher the genus, the more twisted the curve. The genus of a curve can be measured by using the Pluckers formula.

Definition 4.1: (Pluckers formula) Let f(x, y) ε k[x, y] be a polynomial of degree d such that Curve C is a non-singular, then the genus of the curve is defined to be

$$g = (d-1)(d-2)/2.$$

Let C be a non-singular curve of genus g over a field F_q. The Jacobian of the curve C is an abelian variety Jac (C) of dimension g defined over F_q. The genus of elliptic curve is 1.

c. Generator and Parity check matrices

We have discussed in section 1.4 about set of rational functions and points [3, 4] on the elliptic curve. We are having an (n, k, d) code and generator matrix is of order k x n and rank of the matrix is k. This means that rows of the matrix are linearly independent. We can generate matrix using k-rational functions and n-points on the curve. Format of generator matrix is

$$\begin{pmatrix} F_1(P_1) \ldots\ldots\ldots\ldots F_1(P_n) \\ \\ \\ \\ F_k(P_1) \ldots\ldots\ldots\ldots F_k(P_n) \end{pmatrix}$$

Next we will see an example for generating generator matrix using elliptic curve. Let our curve E : $y^2 = x^3 + 5x + 4$ over a field F_7. Here O is the point in infinity and there are 8 points. Points are computed using point counting program in Appendix –B IV.

$P_0 = (3,2)$	$P_3 = (0,5)$	$P_6 = (4,5)$
$P_1 = (2,6)$	$P_4 = (5,0)$	$P_7 = (2,1)$
$P_2 = (4,2)$	$P_5 = (0,2)$	$P_8 = (3,5)$

So here n=9, Let the vector space is defined over L(4P) , then k = 4 and d = 5 there by (n , k, d) code here is (9,4,5) code.

Let the rational functions be $F_1 = 1$; $F_2 = x/z$; $F_3 = x^2/z$ and $F_4 = xy/z$, from these we can generate a generator matrix for a given code. Generator matrix is of order k x n, where k is the set of rational functions and n is the number of points. After computing rational functions, the pole orders of point O on those functions are computed and functions are selected on the order of their values and not selecting two functions with same values. This makes generator matrix linearly independent. The generator matrix here is a 4 X 9 matrix and is as follows

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 2 & 4 & 0 & 5 & 0 & 4 & 2 & 3 \\ 6 & 5 & 1 & 0 & 0 & 0 & 6 & 2 & 8 \\ 2 & 4 & 2 & 0 & 4 & 0 & 2 & 4 & 2 \end{pmatrix}$$

Parity check matrix for a code can be generated from the given Generator matrix. If C is a code and C' is the dual code, generator matrix of C is the parity check matrix of C' and vice-versa. Once matrix is generated, code can be constructed as follows

C = UG, U is the information transmitted of size (1 x k). Some important results of Coding theory [3] are

1. Rows of generator matrix are linearly independent.
2. Columns of parity check matrix are linearly independent
3. $G.H^T = 0$
4. CH = 0. These results are used for decoding

4.1.3 Limitation

In this chapter we are discussing about various requirements needed for developing a cryptographic algorithm using Algebraic Geometric code. Main aim is to generate an algorithm with maximum security. So when we choose the parameters, it should be in such a way that, constrains of security is optimized. Limitation here is when size of field increases computation difficulty increases. But it can be overcome by increasing the capacity of the processor.

4.2 Design of a Cryptographic algorithm using Algebraic Geometric Code

Cryptography is the process of transmission of information over secure channel. In order to retain security of information, we are converting the information into a format that is not easily understandable by an intruder or a person who is illegally trying to acquire the message. As discussed in Chapter two, we are having Symmetric Cryptography and Asymmetric Cryptography [7-9]. Asymmetric cryptography is also called Public key cryptography. In Public key Cryptography there will be two keys –one public key and other private key. Public key is a key known by all people in the network. But private key is a secret key known only to sender and receiver.

4.2.1. Key generation

i. User X select a random integer α , between 0 and ord $_B$, where B is the base point of the chosen elliptic curve.
ii. User Y select a random integer β , between 0 and ord $_B$ where B is the base point of the chosen elliptic curve.
iii. Compute $P=\beta B$.
iv. Public key information includes F_q , P, χ the elliptic curve.

Here we are assuming that data is communicated between user X and user Y. The curve selection is done here by Koblitz random selection method. The method is given in Appendix B-VI.

4.2.2. Encryption

Process of encryption involves the usage of generator matrix G[5] constructed using principles described in section 1.4.

Let M be the message to be transmitted. Group the message into k units i.e. $m_1, m_2 \ldots m_k$. These units of messages are mapped into integers and stored in a vector U.

i. Compute $C = U_{1,k} * G_{k,n} \pmod q$.

ii. Compute the $C^1 = \alpha P.C + Z$ Where Z is a random error vector of length n that introduces t errors which is used for detection and correction of errors. Multiplication of αP to C is done by masking process [10, 11].

iii. Send (C^1, γ) to the receiver, where γ is α B.

4.2.3 Decryption

Receives the pair $[C^1, \gamma]$

i. Multiply β to γ to get α P.

ii. Compute C^{11} from C^1. $C^{11} = C^1 / \alpha.P$.

iii. Compute C^{11}. H where H is the parity check matrix for the code.

If $C^{11}.H\ [3] = 0$ no errors in the information transmitted.

Multiply C^{11} to G^{-1} [12] to get U. The U is converted into M.

Else

Go to 4.2.4.

Proof: The Proof of the functionality of algorithm is as follows. During the process of encryption Code C is formed by message converted into integers and Generator matrix. The keys are selected with the help of the Base point. Multiplication of key information to the contents of code is done by the Masking process. Masking is the process of multiplying (x, y) of key α P to C. By doing this cipher text is not imbedded as points on curve but they are as the contents of field elements [10, 11]. At the receiving end the receiver compute α P and divide it w ith the information received resulting in encoded information. This can be decoded to get the information transmitted. If no errors by taking inverse of G [12] we will get the needed information. This is because of the linearly independent property of Generator matrix. The G here is a asymmetric matrix and left inverse of G is taken.

4.2.4 Decoding

Decoding is done for detection of errors. If no errors, Z component will be zero. So in the decryption step we will receive the code transmitted. Decoding is done as follows

1. Compute the syndrome.

$S = C^{11}.H$ where $C = C + Z$

$C^{11}.H = (C^{11}+Z).H$

$= C^{11}.H + Z.H$

$= 0 + Z.H = Z.H$

We now got an equation to find out errors. Let $Z_1, Z_2 \ldots \ldots Z_k$ be errors.

2. $H(C^{11}) = H \begin{bmatrix} Z_1 \\ \vdots \\ Z_{n-k} \end{bmatrix}$

Here $Z_{1.....}Z_{n-k}$ can be obtained by finding a low weight linear combination of columns of H summing to S. This means we can obtain values of error vectors by solving n linear equations in terms of Z to values of S. So

$U = C^{11} - Z.$

The details of decoding algorithms are given in section 4.4.

4.3 Design of Cryptographic algorithm using the Concepts of Repetition codes

In this session we will see how we can develop a cryptosystem by making use of concepts of Algebraic Geometric code. Main thing in a cryptosystem is the generation of keys.

First, we have to choose some public key parameters. An elliptic curve having a highly secured point over a finite field is chosen along with a fixed base point. The selection of curve is done by using Kobliz selection method given in Appendix B-VI. Public information include

1. Elliptic curve
2. Finite field

Once we know the elliptic curve and field on which the curve is designed we can compute the linear vector space L (D) [5] over the curve in finite field. From the linear vector space we can generate rational functions and also compute base point for the selected curve. From the theory of Algebraic Geometric code we can generate a generator matrix $G_{k \times n}$ by using rational functions and points on selected curve. The rational functions are selected on the strict pole order of the base point B. Next step is the process of generating algorithms. The process involves four steps

1. Key generation
2. Encryption
3. Decryption
4. Decoding

4.3.1 Key generation

There are two types of keys in a public key cryptosystems: - Public key and Private Key. Consider that message is transmitted between two users X and Y, keys can be defined as follows

i. Public key information includes F_q, χ, P Where, F_q is the finite field, χ is the elliptic curve.

ii. User X selects a random integer α, between 0 and ord $_B$, where B is the base point of the chosen elliptic curve.

iii. User Y selects a random integer β, between 0 and ord $_B$ where B is the base point of the chosen elliptic curve.

iv. Compute $P = \beta B$.

4.3.2 Encryption

Encryption is the process of converting a message into a form that is not understandable by a third person. This is done by making use of the keys. Let M be the message to be transmitted. Group the message into k units i.e. m_1, $m_2 \ldots m_k$. Convert this message into points. Conversion of messages into points on curve is called message imbedding [11]. The generator matrix can be constructed as follows. From the divisor D of the curve, find a sub space A and let L (A) be the linear subspace associated with A. Let f_1, $f_2 \ldots$ f_t be the functions related to it. From this, we can generate a generator matrix by using L (A) rational functions and message converted points.

E: L (A) $\rightarrow F_q$

F$\rightarrow$ $((f_1 (p_1), f_2(p_2)\ldots \ldots f_1(p_k))$

Let G be the generator matrix created. Encryption process involves following steps.

i. Compute $G^1 = [G_{i*k} + \alpha P], \gamma]$ where $\gamma = \alpha.B$.
ii. Send G^1 to Y.

4.3.3 Decryption

Decryption is done at receiving end to convert data into its original form. The process includes following step

i. Compute α P.
ii. Subtract α P from G^1.
iii. By taking rational functions and solving them, we will get the points represented through generator matrix.
iv. Points are then converted into messages units and they are in turn converted into original message.

4.3.4 Decoding

Decoding process includes the process of error detection and correction. Here, we are sending information as contents of generator matrix. When we analyze it, we can see that it is a repetition of the point information, there by we can treat it as repetition codes. The simplest kind of error detection is done by making use of repetition codes .When we solve step 3 of decryption algorithm we will get a set of repeated information. Every data is repeated around at least t times. If they are repeated t times, we can say there are no errors. Otherwise it can be assumed that error has occurred. Once an error is detected we can select a point that is repeating maximum number of times. A drawback of repetition code is redundancy, which means we have to transmit

more information to achieve the required result. Although the process seems to be cumbersome, it is simpler than other cryptosystems using algebraic geometric code.

4.4 Design and study of decoding algorithms

4.4.1 Introduction

The construction and decoding of Algebraic Geometric codes are two important tasks in the development of algebraic geometric code. When ever we construct a code it should be possible to correct maximum number of errors. Decoding is done for the process of detecting errors that were accumulated through an unreliable channel. Main aim of decoding is to reconstruct the original code word from its corrupted form.

Generator matrix G and parity check matrix H are two main concepts used in the construction of Algebraic Geometric code. The important property used here is linearly independent property of the rows and columns of these matrices. For a given Algebraic Geometric code C= mG , the important property used in the process of decoding is C.H = 0[5]. Most of the decoding algorithm uses the technique of solving systems of linear equations which is purely dependent on linear algebra.

Given a received pattern, the main aim of decoding process is to decide what the transmitted code is. The decoder tries to find the error pattern e by assuming code word received as C = r +e, to find e, the following formula is used

$$H.r^t = HC^t + He^t$$
$$= 0 + He^t$$
$$H.r^t = He^t$$

This product is called the syndrome and it helps to reveal the error pattern in the received word If a single error has occurred during the transmission, the error pattern will have single 1 in the bit position in which error has occurred and zero in other positions.

There are many decoding algorithms available. But complexity is very high so it is practically difficult to implement it. Most of the decoding algorithms can correct up to (d-1)/2 error that occurred during the process of communication. Various decoding methods include maximum likelihood decoding, majority voting scheme, decoding using displacement scheme, decoding using key equation etc.

4.4.2 Decoding algorithms

The algorithm that is used to decode a given code are called a decoding algorithm. Various decoding algorithms are available. Each algorithm tries to correct maximum number of errors. Driencourt [13] was the first person who approached to correct Algebraic Geometric code ,but it could correct only very few number of errors. Later Justesen, Elbrond Jensen, Havemose and Hohold [14] found a generalization of the decoding algorithm developed by Arimoto[15] and Peterson for Reed Solomon codes to Algebraic Geometric code over plane curves. Various decoding algorithms available include decoding using key equation, list decoding, majority decoding, Duursma and Breeklekamp-Massey algorithm [16]. Most of the algorithms can detect up to (d-1)/2 errors .In this thesis a very few algorithms have been discussed and then a n approach to utilize it in the cryptosystem using Algebraic Geometric code on elliptic curves is made. It is done in such a way that maximum numbers of errors can be detected and corrected. First we will start with a basic decoding algorithm that can be modified and used in this thesis.

a. S-V algorithm

It is one of the first methods of decoding Algebraic Geometric code. It computes error locator, error locator polynomial and computation of error values. It can also be called as error locator decoding. An error word is defined as $(e_1 \ldots e_n)$. For an error word e_i the point P_i is called an error location if $e_i \neq 0$. An error locator exist for an error word e in linear vector space L(A) if and only if t is the weight of e. Let A be a divisor with support distinct from D, if dim (A) > t and it is defined as a function

$$\theta = b_1 \sigma_1 + \ldots\ldots + b_s \theta_s \text{ iff } b_1 \sigma_1(P_1) + \ldots\ldots + b_s \sigma_s(P_1) = 0$$

for all error P_1 . This can be formulated as equations of s unknowns. This polynomial is called error locator polynomial.

The SV algorithm is Skorobogato-Vladut [14]algorithm that can be used for decoding a dual code $C_\Omega(D,G)$, with n = deg(D) and n > deg(G) > 2g-2. To define this algorithm we choose 3 divisors A,B and C so that $B \subset G$ and A + C > D. Error correction capability of this code is $\lfloor (d-1-g)/2) \rfloor$. Let the code $C_\Omega(D,G)$ defined over a curve of genus g over a field F_q , Let $\theta_1 \ldots\ldots \theta_k$ be a basis of L(G).It can be defined by evaluation map at P as

$$\theta : L(G) \rightarrow F_q$$

Let D be of the form $D = P_1 + \ldots\ldots + P_k$. Let basis of L(A), L(B) and L(C) can be represented by

$$L(A) = \text{span } \{ \theta_1, \theta_2 \ldots\ldots \}$$

$$L(B) = \text{span } \{ \lambda_1, \lambda_2 \ldots\ldots\ldots \}$$

$$L(C) = \text{span } \{ \varphi_1, \varphi_2 \ldots\ldots\ldots \}$$

The decoding can be done as follows

1. Compute the syndrome S as follows

$$S = \begin{bmatrix} w.\theta_1.\lambda_1 & w.\theta_1.\lambda_1 \ldots\ldots\ldots\ldots \\ w.\theta_n.\lambda_n & w.\theta_n.\lambda_n \ldots\ldots\ldots\ldots \end{bmatrix}$$

$$= \text{W.A.C.}$$

If all syndromes are zero then the received word is the cord word.

2.Find a non-zero vector ($b_1, b_2. \ldots.$) in the null space of S.

$\theta = b_1\theta_1 + b_2\theta_2 + \ldots..$ is an error locator.

3. Find error locations M such that $M = \{ 1 \leq i \leq n \mid \theta(P_i)=0 \}$.

4. Find the error values e_i by solving the linear system

$$\Sigma_{1 \in M} \varphi_1(P_1)e_1 = W.\varphi_i .$$

Solving this linear equation we get errors occurred and the original data can be obtained by C = W - e . Major draw back of S-V algorithm is that it does not have full error correction capability of code.

b. Duursma Algorithm[17]

Let C be a code word C(X, D, G), based on the curve X of genus g and let f = C + e be the received word, where $wt(e) \leq t$ and deg $G \geq 2t + 2g - 1$.The procedure for the decoding is as follows . Choose a divisor A with deg A = t and support disjoint from one of D. Find $G^1 = G - (\deg G + 2g + 2t - 1)Q$ and $A^1 = G^1 - A - (2g-1)Q$. Choose basis $\{\varphi_0, \varphi_1, \ldots \varphi_{3g+2t-1}\}$ of $L(A + A^1)Q$, $\{\psi_0 \ldots \psi_{t-1}\}$ of $L(A+(3g-1)Q)$ and $\{\chi_0, \ldots \chi_{2g-t-1}\}$ of $L(A^1 + (2g-1)Q)$, indexed respectively by the $(A + A^1)$, A and A^1 orders.

1. Syndrome matrix can be calculated as follows.

   $S_{ij} = \psi_i \chi_j . f$. If the syndrome is zero no errors occurred.

2. Find the error locator

Let u be the maximal order of the known rows. Look for a non-zero solution of the linear system $\sum_{i \le u} \alpha_i S_i = 0$ where S_i is the i-th row of S. If a solution α exists, then $\sum_{i \le u} \alpha_i S_{\perp} = 0$ is an error locator, go to step 5.

Let u be the maximal order of the known columns. Look for a non-zero solution of the linear system $\sum_{i \le u} \beta_i S_i = 0$ where S_j is the j-th column of S. If a solution S_j *exists* then is $\sum_{i \le u} \beta_i S_i = 0$ an error locator: go to step 5.

3. Estimate additional syndromes

Assume that every entry of order s_j is known, while no entry of order s is known. For each pair (r, r') with $r + r' = s$, try to solve the linear systems

$\sum_{i<r} S_{i,k} \alpha_i = -S_{r,k}$ for $k < r'$

$\sum_{j<r'} S_{r,j} \beta_j = -S_{r,r'}$ for $k < r$

4. Majority voting

For each test entry $S_{rr'}$ use the expressions of $\psi_r \chi_{r'}$ in terms of the basis φ_i, and the known syndromes $\varphi_i e$ for $i < s$, to calculate the vote $\varphi_i e$. The true value $s \pm e$ is the vote that occurs most frequently. Using this value, recalculate all the syndromes $\psi_r \chi_{r'}$ (all but the test entries that gave correct votes). If an additional column or row is known, go to step 2, otherwise go to step 3.

5. Find error locations: Using error locator φ, determine

$$M=\{P_i \varepsilon \text{ supp}(D) \mid \varphi(P_i)=0\}$$

This is a set of error locations.

6. Error values can be calculated as follows

$$\sum \varphi(P_l) e_l = \varphi_k \,.\, k \quad \text{for } k \varepsilon\, u + 2g + t - 1.$$

$P \varepsilon M_1$

Main difference between the Duursama and S-V algorithms is majority voting. Because of that more errors can be corrected in Durrsama algorithm then S-V algorithm.

c. List decoding of Algebraic Geometric code

List decoding of Algebraic Geometric code [18] can decode Algebraic Geometric code beyond conventional error correction bound (d-1)/2. This method involves two steps, factorization and interpolation of polynomials over finite fields. The basic result of list decoding is as follows.

Theorem 4.1: Let C be an AG-code of block length n, dimension k, Field F_q, curve X of genus g. Then, for any positive integer b, C is (n-β-1, b)-decodable, where

$b := \lfloor (n+1)/(b+1) + b\alpha/2 + g - 1) \rfloor$ and $\alpha = k+g-1$.

The principle followed in list decoding is to compute a list of at most n code words ,one of them must be x

1. Interpolation step:- for a non zero polynomial $H(T) = u_b T^b + \ldots + u_1 T + u_0 \varepsilon K[T]$,where $u_j \varepsilon L(F + (b-j)G$, such that $H(P_i, y_i) = \sum_{j=0}^{b} u_j(P_i) y_i^j$ is zero for i = 1,...n.

2. Factorization step: Find all roots of H(T) in K. For each root ρ compute $x_p = (\rho(P_1), \ldots, \rho(P_n))$, if x_p is not defined or if distance between x_p and y is larger than n- β-1, discard x_p.

d. Displacement approach of decoding

This method uses a method of displacement [17] for efficiently computing a nontrivial element in the kernel of a structured matrix. The method can be briefly described as follows. Let X be a curve and L (αQ) be the

vector space associated with the function field, Assume that decoding code is of length l. Let $\beta = \lfloor (n+1)/(l+1) + l\alpha/2 + g \rfloor$. Let $\varphi_1, \ldots, \varphi_t$, $t = \beta - g + 1$, be the elements of $L(\beta Q)$ with strictly increasing pole orders at Q. Let $(y_1 \ldots y_n)$ be the received word, the non-zero element in the kernel of the matrix can be found as

V =

$$\begin{pmatrix} \varphi_i(P_1) \ldots\ldots \varphi_{s0}(P_1) & y_1\varphi_i(P_1) \ldots\ldots y_1\varphi_{s0}(P_1) & y_1^{\,l}\varphi_i(P_1) \ldots\ldots y_1^{\,l}\varphi_{s0}(P_1) \\ \varphi_i(P_2) \ldots\ldots \varphi_{s0}(P_2) & y_2\varphi_i(P_2) \ldots\ldots y_2\varphi_{s0}(P_2) & y_2^{\,l}\varphi_i(P_2) \ldots\ldots y_2^{\,l}\varphi_{s0}(P_2) \\ \varphi_i(P_n) \ldots\ldots \varphi_{s0}(P_n) & y_n\varphi_i(P_n) \ldots\ldots y_n\varphi_{s0}(P_n) & y_n^{\,l}\varphi_i(P_n) \ldots\ldots y_n^{\,l}\varphi_{s0}(P_n) \end{pmatrix}$$

With the help of a diagonal matrix diag $[\varphi]_{l,n}$ and an upper shift matrix A we can form a displacement matrix DV – VZ = GB, where $G \varepsilon F_q^{\,n \times dl}$ and matrix $B \varepsilon F_q^{\,dl \times n}$.

e. Modified decoding algorithm for elliptic curves

Here we have seen various algorithms for decoding algorithms. Here is a modified algorithm that can be used to detect maximum number of error [19, 20, 21]. Principles followed here is similar to S-V algorithm. The steps involved in the process are as follows.

Let X be a curve, P be a set of points on curve, Q be the base point on curve and D be the divisors on the curve and code is represented as (n, k,d)and L(D) represent the linear vector space of the curve and L(D)=span$\{\varphi_1, \varphi_2 \ldots \varphi_k\}$. All vector space rational functions are strictly based on the increasing pole order values at Q.

1. Compute the syndrome matrix by using vector space L(A) and L(B) where L(A) = span $\{\psi_1, \psi_2 \ldots\ldots \psi_m\}$ and L(B)= span $\{\chi_1, \chi_2 \ldots\ldots \chi_m\}$

S = M. A. B

If S = 0 , no errors occurred.

2. i =1

3. Compute error locator polynomial $\theta = b_1\varphi_1+.....+ b_m \varphi_m$ such that

$$b_1 \varphi_i(P_1) + \ldots + b_m \varphi_m(P_1) = 0 \quad (4.1)$$

$$\vdots$$

$$b_1 \varphi_i(P_n) + \ldots + b_m \varphi_m(P_n) = 0 \quad (4.2)$$

By solving equations (4.1) through (4.2) values of $b_1 \ldots b_m$ can be obtained. If no solution exists compute i = i+1 and repeat the process until a solution till m + i = k.

5.Find the error locations E such that $E = \{ 1 \leq i \leq n \mid \theta(P_i) = 0 \}$

6. Find error values e_i by solving

$$\sum_{i \in E} \varphi_i(P_i)e_i = M.\ \psi_i \quad (4.3)$$

7. M = M – e.

Repeat the step 2 to 5 by incrementing m in equation (4.1) to k by 1. If we do this we can extend (d-1)/2 to a maximum level.

4.5 Design of Secret Sharing algorithms

Secret sharing as discussed in the chapter 3 is the method of sharing a secret among many users. The share can be recovered only by a predetermined set of users. Here a secret sharing method is used in which secret is splited and distributed among various users. The secret selected is depended on the curve and algebraic code parameters. The process involves many steps. Main process is finding a set of authorized users. It should be coordinated by a dealer or an administrator. Secret should be shared only to the authorized users determined by the dealer. Secret sharing involves three steps

a. Secret splitting or set-up phase

This phase can also be called secret splitting phase. The steps involved in the process are as follows. Since the cryptographic algorithm used here is dependent on Agebraic Geometric code and elliptic curve, parameters should be dependent on the elliptic curves. The steps are as follows.

i. Generate a curve E with a field size of sufficiently large prime p.

ii. Compute a base point for E and order n of the base point.

iii. Dealer generates a random number r with a limit to n.

iv. Split r to $r_1, r_2 \ldots r_m$.

v. Generate a polynomial $F(x) = r_1 x^m . r_2 x^{m-1} + \ldots\ldots + C \pmod q$.

Where C is a constant generated by the dealer.

b. Secret distribution

This step involves distributing secret to m users. This is done with the help of polynomial generated in secret set up phase.

i. Compute $f(i)_{i=1 \text{ to } m} = F(i)_{i=1 \text{ to } m}$

ii. Distribute values of f(i) to i^{th} user.

c. Secret reconstruction

After construction of secret, information is encrypted and at the receiving end dealer collects the secret information from its authorized user. Let $a_1, a_2 \ldots\ldots a_j$ be the secret information and let it be represented in an array f then

1) $F(i)_{i=1 \text{ to } m} . r_{i=1 \text{ to } m} = f_{i=1 \text{ to } m}$

2) $r_{i=1 \text{ to } m} = F^{-1}{}_{i=1 \text{ to } m} \cdot f_{i=1 \text{ to } m}$.

3) Compute secret $s = r_1 + r_2 + \ldots + r_m + c$.

Thus secret is reconstructed and can be used to decrypt the information received.

4.6 A Digital Signature for the System

4.6.1 Introduction

Digital signature is the subset of electronic signatures that make use of the concept of Cryptography. Digital signature tries to combine the signature in real world taking into account properties of the electronic world. If electronic mail system are to replace the existing paper mail system for transactions, signing of message is necessary. It is a process of signing the document so that later the problem of authentication of the documentation will not arise. It is an analogue of handwritten signature. It was evolved from what is known as electronic signature. Electronic signature can be defined as electronic sound or process, attached to or logically associated with a contract or other record and executed or adopted by a person with the intent to sign the record. [22,27]. The signature doesn't depend on identity of the signer (private key) but also the information that is being transmitted. The properties that are provided and assured by the use of digital signatures include

- Authenticity: - Verifier after successful verification should be assured that the Information was signed by the provider of the digital signature.

- Integrity: - Both sender and the receiver of the signed message shall be confident that a message has not been modified by an intruder.

- Non-repudiation:-Signature can be shown to make them accepting the ownership. There is a possibility that when a system got broken, the author of the message deny the ownership of the message. This can be avoided by making use of digital signature.

The recipient of a signed message takes it as a proof of the message originated from sender. A digital signature must be a message dependent, as well as signer dependent. Digital signatures are created and verified by cryptography. The process of creating a digital signature and verifying it, accomplish the essential efforts and can be used for many legal purposes. The authentication property described above in digital signature is of two types. Signer authentication and Message authentication. [22, 23]

- Signer authentication: - If a private and public key pair is associated with an identified signer, and the digital signature attributes to the signer. The signature cannot be forged, unless signer loses the control of the private key.

- Affirmative act: - Creating a digital signature requires signing a private key. This act can perform the ceremonial function of alerting the signer to the fact that signer is performing transaction with legal sequences.

- Message authentication: - The digital signature also identifies the signed message typically with far greater certainty and precision than paper signatures. Verification reveals only tampering, since the process involves the process of hashing , which shows the message is same as signed.

The process used for digital signatures have undergone technological performance test over a decade. Digital signatures are accepted in several national and international standards and accepted by much cooperation, bank and government agencies.

A general public key digital signature algorithm [25,26] involve the following step

a. Key generation
b. Signature generation.
c. Signature verification

The digital signature process can be diagrammatically described as shown below.

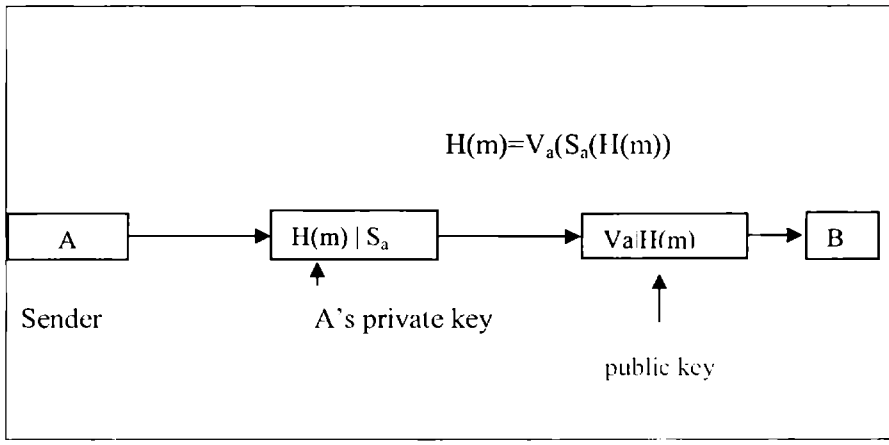


Fig 4.1 A Digital signature scheme

A message m is digitally signed and it is sent to receiver. The receiver checks the signature and if matching accepted. Digital signatures use hash function to do its processes. In case of long message, the signing and verification process may be very time consuming. The idea is to sign smaller amount of information without compromising the system security.

4.6.2 Cryptographic hash functions

A Cryptographic hash function that takes output of arbitrary length and it convert it to fixed length. Cryptographic hash function produces digest (finger print) from an electronic document usually much shorter than original document. A hash function is usually a projection

H: x→y where y is a finite set and x can but (doesn't need to be) be a finite set. Value x ε X is called document message; value h(x) is called digest. Value of H(x) can be used as substitute of original document x. Hash functions can be based on various principles. It includes NP hard problems, modified block ciphers or dedicated hash functions that can be designed or reused. Cryptographic hash functions are functions which have many uses in

cryptography. It can play an important role in proving the security of public key signature schemes and /or public key encryption schemes and key agreement protocols [27,28].

A hash function to be used in cryptography has to satisfy following requirements.

- Preimage resistance: - Given a hash value h, it is impossible in practice to find a message m with H (m) =h.
- Collision resistance: - It is impossible in practice to find message m and m^1 with $m \neq m^1$ and H (m) = H (m^1)

Currently, hash values of bit length N = 160 are considered to be sufficient in general for a hash function to be cryptographically strong. Certain hash functions are as follows

Hash Algorithm	Hash sum size(bits)
MD2	128
MD4	128
MD5	128
SHA-0	160
SHA-224	224
SHA-512	512

Table: 4.1 Hash algorithm and bit size

MD5 is an iterated hash function introduced by Ronald Rivest in 1991 as a successor to MD4 and become internet standards RFC 1321, ensuring its widespread occurrence in many contemporized applications and standard. SHA (Secured Hash Algorithm) is a class of iterated hash functions. Various version of SHA are available as specified in the above table.

A modified version of digital signature is as follows

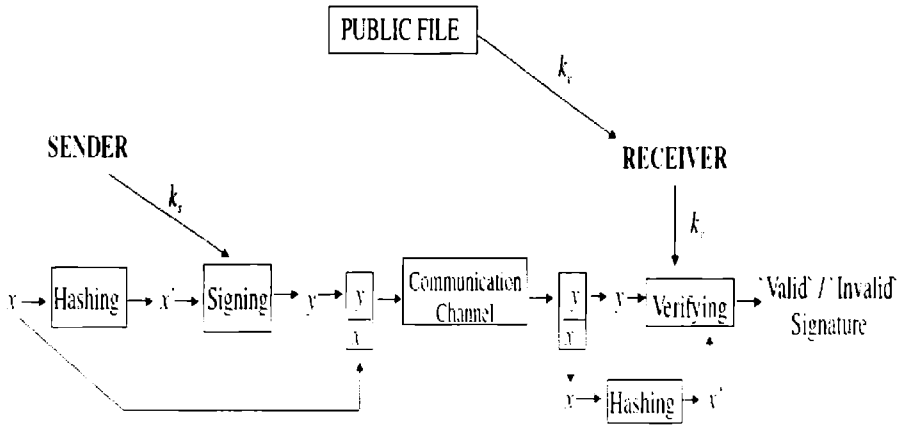


Fig 4.2[22] Functioning of a signature system

Here the signature system is signed with the help of a hashing function and at the receiving end the signature is verified and message accepted only if it is a valid one.

4.6.3 Proposed system

In this section we will see how digital signature can be applied to cryptosystem based on Algebraic Geometric curves and codes [24]. The algorithm for key generation, signature generation and verification is as follows.

4.6.3.1 Key Generation

Key generation is the process of generating key for the process of signature generation. The parameters of elliptic curve, finite field and Base point of the curve is necessary for the generation of public key. Private Key involve a random integer that depend on the parameters of elliptic curve . The procedure for the key generation is as follows.

Each entity create a public key and a private key

i. User X select parameters [q, E, a, b] of the curve, q should be a large prime field, E an elliptic curve, a and b are parameters of E.
ii. Compute a base point B for the curve E.
iii. Generate a random integer l such that $1< l < \text{ord}_B$. A's public key is [E,Q = lB].

4. 6.3.2 Signature Generation

Signature generation involve selecting a random integer and computing the signature using it. The generated signature is sent to the receiver along with the message and key parameters. The signature generation involves usage of hash function. Here in this procedure message is not sent to the hash function. But the encoded information is sent to the hash function. This is done to increase the security of the system.

User X should do the following

i. Select a random integer k such that $1< k <\text{ord}_B$.
ii. Compute C= m.G,
    P = kB,
    E = H(C)
iii. Compute $S = k^{-1} [E - Q_x(\text{mod } q)]$
iv. User X send the pair [P, S] to Y

4.6.3.3 Signature Verification

The receiver on other hand on receiving the information transmitted, verification is done. The information he got is accepted only if the verification process is successful. To verify X's signature on message m, Y should do the following.

User Y should do the following:-

Given [P, S]

i. Compute E=H(c)

ii. Compute V1 = (S.P +B Q_x) mod q.

iii. Compute H(C) and if (H(C) != E) "invalid".

iv. V2 = B.H(C) mod q.

v. Accept the signature if V1 = = V2.

Proof: Received information contains signature S = k^{-1} [H(C) – Q_x]mod q, , E=H(C) P = kB. Q_x is the x coordinate of Q .User Y will do the following. V1= (BQ_x + k^{-1} [k .B.H(c) –kBQ_x]) mod q i.e. V1= B.H(C) mod q. V2 = B.H(C) mod q. If V1 = V2 , we can accept signature other wise reject it .

4.6.3.4 Security aspects of the Digital Signature Algorithm

The security of the system can be discussed as follows.

a. An adversary might attempt to forge A's signature of message m by selecting random integer k and then determine s = k^{-1}(H(C) – Q_x) mod q. If elliptic curve desecrate logarithm problem is computationally infeasible, the adversary can succeed with a success probability of 1/p which is infeasible for a large prime no. So system can be accepted only if we are using a large prime number. Since each message is signed with a new random number finding k is infeasible.

b. To implement sign, hashing function is necessary otherwise the adversary can easily find the signing parameters by mapping one content to another.

c. Security is based on the selection of parameters also. Parameter selection includes curve, size of field and private key selected. The

field size should be sufficiently large and Order of base point P should be divisible by a large prime.

4.7 Security issues of the Cryptosystem using Algebraic Geometric Codes

Computers and electronic media are widely used for transferring sensitive information, plays a vital role in the area of communication. There comes the concept of cryptography where information's are encrypted and sent. But there is a possibility that cryptanalyst or an intruder try to break the system. So even when we develop a cryptosystems, security of the cryptosystem is very important. It means cryptosystem should not be prone to attacks. We are concerned with security of a cryptosystem developed using Algebraic Geometric codes. Here Algebraic Geometric code is developed by using elliptic curve and there by certain concepts of Elliptic curve cryptography is also used. Mc-Eliece developed a Cryptosystem based on codes. Although it was secured from all attacks, its key size was very large. Due to this reason it is not used extensively. In this chapter, a study is done on the effect of various attacks on the cryptosystem developed using the concepts of Algebraic Geometric codes and elliptic curves.

4.7.1 Introduction

Security is the planning, implementation and enforcement of a series of policies which can be transmitted via communication channels by guarding against threats. Security of a cryptosystem is evaluated by amount of time needed to break it. Here breaking means finding the private key used for encryption and getting the information transmitted. The process of breaking the system is also known as attack methods of the system. Amount of time required to break a cryptosystem is a theoretical estimate of average time needed to break a cryptosystem by a given attacking method.

In this section we deal with how the attacks affect the performance of a cryptosystem developed using algebraic geometric code. Also we can see how to get optimized parameters for generation of cryptosystem developed using Algebraic Geometric codes based on elliptic curve, where key parameters are chosen from elliptic curves.

Above described cryptosystems involve key generation, encryption and decryption. When we are developing a system using Algebraic Geometric code we make use of generator matrix [5] and a private key which depends on a random integer generated using the concepts of elliptic curve.

One another advantage of the cryptosystem described above is the decoding. Once key is retrieved, the form of message is an Algebraic Geometric code. The code can be decoded to find the errors that had occurred during the process of transmission. Information is transmitted via communication channel. Channels are always prone to errors due to noise and other disturbances. So apart from an intruder, the information transmitted should be protected from the channel errors also. By making use of a decoding algorithm we can detect up to (d-1)/2 errors, where d is the dimension of the code transmitted.

Here this crypt analysis depends on two factors. First, cryptanalyst should get the private key that depends on the parameters of the curve. The second depends on the structure of the generator matrix developed.

4.7.2 Attacks on the cryptosystem

The security of the cryptosystem is computed by the amount of time taken to break the system. The cryptanalyst's main aim is to find the key used in the encryption algorithm. The method used to break the cryptosystem as mentioned above is called attacking method.

The algorithm discussed above uses concepts of Algebraic Geometric code and Elliptic Curve cryptography. When attacks are taken into consideration, main issue here is the private key generated which is dependent on the elliptic curve. So we can say that the key concepts here are similar to elliptic curve cryptography. Next section discusses various attacks common to cryptosystem using algebraic geometric code and elliptic curve cryptography and how these attacks affect the system. First the attack on Algebraic Geometric code [29, 30] is discussed and in the next section elliptic curve based attacks are considered.

a. Known partial plain text attack

Having partial knowledge of the plaintext drastically reduces the computational cost of cryptosystem. For example, let $m_1,m_2....m_n$ be the message received and if the intruder knows $m_1,m_2...m_k$, it is easy to reveal the key and there by getting the information transmitted, provided, they know the structure of generator matrix. Here is the advantage of usage of coding concepts in cryptography. Even if the intruder know the partial message, it will be impossible to lay his hands on the entire message. He needs to have knowledge of the generator matrix used in the process.

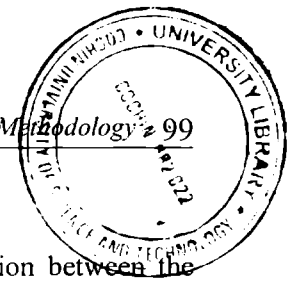
b. Message resend attack

Let the information transmitted is the form $c=\alpha mG$, where, m is the message transmitted and G is the generator matrix [2] Consider the situation that the message is sent again. The crypt analyst who is trying to reveal the message will now have

$$c1= \alpha mG$$

$$c2= \beta mG$$

Since each time the system is using a new random integer, attack is as difficult as revealing a newly sent message.



c. Related message attack

Sometimes the cryptanalyst knows some linear relation between the messages sent. This is called related message condition. This is possible only if the private key is known. When two cryptograms C_1 and C_2 are combined it will be

$$C_1+C_2 = \alpha m_1 G + \beta m_2 G$$

Since each time different key is generated, it is not possible to combine the messages simply by knowing the relationship between messages.

d. Information set decoding

This is a method by which once a cryptanalyst get the information transmitted, he will try to get the message by applying any known decoding algorithm by randomly choosing n, k parameters, i.e. subset of generator matrix. Once he got the decoding information, he can again randomly apply some private keys and he may succeed in getting message transmitted. The average amount of work performed is proportional to the number of operation required in decoding and generating key.Of the known general attacks (i.e., not against specific codes etc.) this seems to have the lowest complexity. One tries to recover the k information symbols as follows: The first step is to pick k of the n coordinates randomly in the hope that none of the k is in error. We then try to recover the message by solving the k × k linear system (binary or over Fq). Let c_k and z_k denote the k columns picked from $G_k^{'}$, c and z, respectively. They have the following relationship

$$c_k = m\, G^{'}{}_k + z_k.$$

If $z_k = 0$ and $G^{'}{}_k$ is non-singular, m can be recovered by $m = c_k\, G^{'}{}_k$. Even if $z_k \neq 0$, m can be recovered by guessing z_k among small Hamming

weights [9] (this is called the generalized information-set-decoding (GISD) attack). One iteration of the algorithm is as follows:

1. Permute the columns of the generator matrix randomly.
2. Apply Gaussian elimination on the rows of the matrix to obtain the form $G = (I_k | A)$,with the corresponding permuted cipher text $c = (c1 + e1|c2 + e2)$.
3. Guess that the error e1 is of weight at most p and check whether the error $e = (e1|e2)$ is of weight t.

Here, in this system mere decoding will not give the exact information transmitted. Again we have get the secret key. Only then we can retrieve the information transmitted.

e. Structural attack

Structural attack is the process of getting the structure of generator matrix used in the process of encryption. Public key information includes the curve and field size. From this, it is easy to generate the structure of Generator matrix. Once we get the structure we can easily create generator matrix. But here, by merely getting G , we cannot resolve the problem. It involves another level i.e. solution to ECDLP. Once we succeed in that , we will get information transmitted.

From the above discussions we can see that above mentioned attacks has less effect on cryptosystems developed here.

4.7.3. Elliptic curve discrete logarithm problem

ECDLP[3] is elliptic curve discrete logarithm problem. This problem involves finding α, from $P= \alpha B$. There are many types of known attacks, like pollard-rho attack, Index-calculus attack, Pohig-helmann attack etc. Here we will see how these attack is carried out on the security of the system.

a. Pohig-Hellmann attack

Pohig-Hellmann attack [7] algorithm efficiently reduces the computation of l=log $_pQ$ to the computation of discrete logarithm in the prime order sub-groups<p>. It follows that ECDLP in <p> is no longer than ECDLP in its prime subgroups. It reduces the determination of l to l_i modulo p_i^{ei} for each of the prime factors of n. Hence in order to achieve the maximum level of security n should be prime.

$$l \equiv l_1 \bmod p_1^{e1}$$

$$l \equiv l_r \bmod p_r^{er}$$

In order to avoid this attack, one should carefully choose elliptic curve parameters so that order n of P is divisible by a large prime. To make order n divisible by a large prime, field size should be extremely large.

b. Polard-Rho attack

Main idea of Pollard-Rho[7] attack is to find distinct pairs (c^1, d^1) and (c^{11}, d^{11}) of integers modulo n such that $c^1 P + d^1 Q = c^{11} P + d^{11} Q$. Then

$$(c^1-c^{11}) P= (d^1-d^{11}) Q. \qquad (4.4)$$

Hence l= $\log_p Q$ can be obtained by computing l= $(c^1-c^{11})/ (d^1-d^{11})$. Parallelized pollard's rho attack is best known for ECDLP. By making use of M processor ands run algorithm in each processor until any one processor terminates.

Whenever we choose the parameters it should be in such a way that it is infeasible to solve.

c. Index-calculus attack algorithms

Index calculus algorithms are powerful in computing discrete logarithms in some groups including multiplicative group Fq* of a finite field. The Jacobean [7] Jc(Fq) of a hyper elliptic curve C of a high genus g defined over a finite field Fq and the class group of an imaginary quadratic number field. This process includes lifting a point to another. It is infeasible in the case of elliptic curve, so this method is a failure in ECDLP.

d. Isomorphism attack

Let E be an elliptic curve defined over a finite Fq and let P ε E (Fq) have a prime order n. Let G be a group order n, such that n is prime, <P> and G are both cyclic and hence isomorphic [31] If one can efficiently compute isomorphism

$$\psi :< P> \rightarrow G \qquad (4.5)$$

Then ECDLP instances in <P> should be sufficiently reduced to instances of the DLP in G namely, given a point P and Q ε <P>. We have $\log_P Q = \log_\psi(P)\ _\psi(Q)$.

Isomorphism attacks[31-34] reduce the ECDLP to DLP in groups G for which sub exponential time or faster algorithms are known. Isomorphic attacks on prime field anomalies reduces ECDLP in an elliptic curve of order p defined over the prime fields Fp to the DLP in the additive integer modulo.

In the case gcd (n ,q)=1 the Weil and Tate pairing[7]attacks establishes an isomorphism between <P> and a subgroup of order n of the multiplicative group F_q^k of some extension field F_q^k. Weil descent attack attempts to reduce ECDLP in an elliptic curve defined over binary field F_2^m to DLP in the Jacobian of a hyper elliptic curve define over F_2^m.Sub-exponential algorithms are available for these attacks.

In 1991 Menzes, Okamoto and Vanstone(MOV) [7] also showed that ECDLP can be reduced to extension field of F_q. MOV is efficient only for special class of curves called super singular curve. Anomalous curves are also not secure curve. In anomalous curve also ECDLP can be easily converted into DLP.

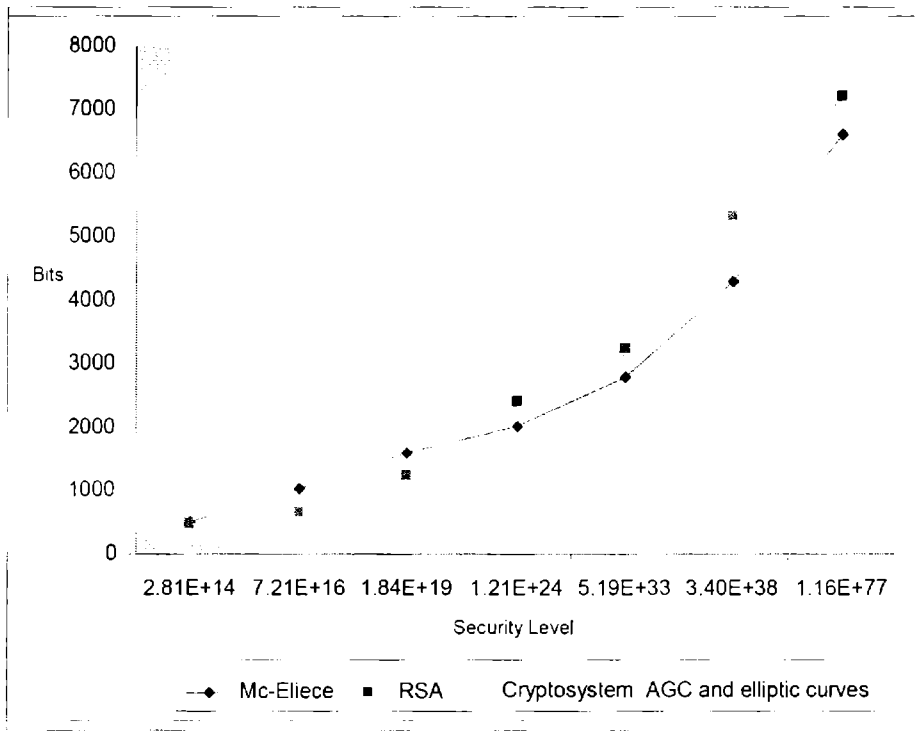


Fig 4.3 Security level vs bit size

The Fig 4.3 shows the security level of Mc-eliece system, RSA system and the proposed system[31-34]. The above figure shows that higher level of security can be achieved for lower bits in the proposed system compared to RSA and Mc-Eliece

It should be noted from the above discussion that the attack involves revealing a single thing i.e. the users private key and is dependent on two factors curve and field size. It can be shown that choice of underlying curve,

representation of elements in the field and field size play a prominent role in the security of cryptosystem using elliptic curves.

First we will consider choosing of the curves. Curves can be selected by using random method, complex multiplication method or Kobliz method [7] When selecting curve for developing a cryptographic algorithm following things should be noted.

- The curve should not be an anomalous curve.
- It should not be a super singular elliptic curve.
- Mathematical operations can be easily performed on the curve

Thereby when we implement algorithm, care should be taken to avoid curves of above mentioned properties [31].

The field also should be carefully chosen to avoid attacks. So while selecting field following things should be noted.

- Field should be sufficiently large prime.
- Order of base point P should be divisible by a large prime.
- Compute gcd (n, q), Where n is the number of points on the curve and q is the size of the field. Avoid field that satisfy gcd (n, q) = 1 because Wiel and Tate pairing attacks can be easily done on the curve of that size.

4.7.4 Conclusion

The above section shows security level of cryptosystem using Algebraic Geometric code developed based Elliptic curve. Various attacks like known partial plain text attack, message resend attack, Polard-rho attack, Pohig-Hellman attack, Isomorphic attacks are seen. It is shown that by carefully choosing the parameters we like field size and curve we can increase the security level. Certain curves like anomalous curves, sub-field curves, and

super singular curves should be avoided. Field size should be carefully chosen and should be sufficiently large prime. There by carefully choosing the parameters we can increase the security level of the system.

4.8 References

[1] Jacobson, Nathan , "Basic Algebra 1",(Second edition), NewYork ,W.H Freeman & Co ISBN 978-0-7167-1480,1985.

[2] R. Liedl, H.Niederreater, "Finite fields, Mathematics and its Applications", Vol. 20, Cambridge University Press 1984.

[3] D.R Hankerson, D. Hoffman, D.A Leonard, C.C Linder, T.T Phelps, C.A Rodger, J.R Wall, "The Coding theory and Cryptography The essentials", (2004).

[4] F.J Mac Williams and N.J.A Sloane, "Theory of Error Correcting Codes", Elsevier publishers B.V, New York, 1997.

[5] V.D Goppa ," Codes on algebraic curves", Sov. Math, Dokl,, pp.207-214, 1981.

[6] Hart Shorne , "Algebraic Geometry " ,Graduate text in Mathematics Vol 52, Springer -Verlag.

[7] Hankerson, Menzes, Vanstone, "Guide to Elliptic Curve Cryptography", Springer. CRC press, 2004

[8]. B.Schneir ,"Applied Cryptography", 1996, Second edition ,Wiley.

[9]. V. Miller, "Uses of elliptic curve in cryptology", Proceedings of crypto'85 LNCS 218, pp 417-426, New York: Springer-Verlag 1986.

[10] A. Menzes and S.Vanstone, "Elliptic Curve cryptography's and their implementation", Journal of Cryptology, pp 209-224, 1993.

[11] D.R.Stinson, "Cryptography Theory and practice", CRC Press Inc, 1995.

[12] B.Noblle and J.Daniel, "Applied Linear Algebra ", 3 rd Edition, pp 167-172.

[13] Driencourt.Y, "Some properties of Elliptic codes over a field of Characterstic 2", Proceedings AAECC-3, Grenoble 1985, Lecture Notes in Computer Science 229, pp 185-193,1986.

[14] S. Sakata, J. Justesen, Y. Madelung, H. E. Jensen, T. Høholdt. "Fast Decoding of Algebraic-Geometric Codes up to the designed Minimum Distance". IEEE Transaction on Information Theory, vol 41(5), pp. 1672-1677 ,1995.

[15] Arimoto, S, "Encoding and decoding of p-ary group codes and the correction system", Information Processing in Japan 2, pp. 320-325,1961.

[16] M.Amin Shokrollahi, Hal Wassweman, " Decoding algebraic-geometric codes beyond the error-correction bound", Proc. 30th Annual ACM Symposium on Theory of Computing,May 1998.

[17] I.M Duursma, "Decoding Curves and Cyclic codes", Eindhoven University Techn, Dissertion,1993.

[18] M.Amin Shokrollah, and Vadim Olshesky, "List Decoding of Algebraic – Geometric Codes", IEEE transactions on Information Theory, Vol (45),pp 423-437,1999.

[19] T. Høholdt, R. Pellikaan, "On the Decoding of Algebraic-Geometric Codes",IEEE Transactions on Information Theory, vol. 41(6), pp. 1589-1614 ,1995.

[20] Anatoly Yu. Serebryakov, "Decoding Algebraic-Geometric Codes over Elliptic Curves when the Number of Errors Exceeds Half of the Designed Distance", ISIT 1998, Cambridge, MA, USA, August 16 -August 21.

[21] V.Guruswami and M.Sudan, "Improved Reed-Solomon and Algebraic Geometric code", IEEE transactions on Information Theory, Vol 45, pp.1757-1767, 1999.

[22] R.L Rivest, A.Shamir and L.Adleman, "A method of obtaining Digital Signatures and public key Cryptosystems", Communication of the ACM, Vol 21, pp.120-126, 1978.

[23] J. Stern, "A new identification scheme based on Syndrome Decoding", Advances in Cryptology – CRYPTO '93, vol. 773 of LNCS, pp. 13–21. Springer-Verlag, 1993.

[24] Jacques Stern, "Can one design signature using error correcting codes", Asiacrypt 1994, LNCS-917.

[25] Philippe Gaborit ,"Shorter Keys for code based Cryptography" , September 2004.

[26] J.L. Massey." Some applications of coding theory in Cryptography", Codes and Ciphers: Cryptography and Coding, Essex, England, Formara Ltd , pp 33-47,1995.

[27] T. ElGamal, " A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, Vol 31,pp.469-472 ,1985.

[28] D. Augot, M. Finiasz, and N. Sendrier, "A family of fast syndrome based cryptographic hash function", E. Dawson and S. Vaudenay, editors, Progress in Cryptology - Indocrypt 2005, LNCS,- 3715,pp 64-83. Springer-Verlag, 2005.

[29] V.M Sidelnikov, S.o Shestakov, "On the insecurity of cryptosystem based on generalized reed-solomon codes", Discrete Math; Vol 1,no.4 pp 439-444. 1992.

[30] C. M. Adams and H. Meijer, "Security-related comments regarding McEliece public-key cryptosystems",Advances in Cryptology-CRYPTO' 87, Springer-Verlag, New York (1987) pp. 224–228.

[31] Remarks on security of ECC, A Certicom white paper published sep-1997.

[32] ECRYPT Yearly Report on Algorithms and Key sizes (2004),Document D.SPA.10, March 2005.

[33] Certicom Research, "SEC 2: Recommended Elliptic Curve Domain Parameters", Standards for Efficient Cryptography, Version 1.0, Sep. 2000.

[34] A.Lenstra and E.Verheul, "Selecting Cryptographic Key sizes", Proceedings of PKC 2000, LNCS1751,pp 445-466 Springer-Verlag 2000.

.....ଌ.....

Chapter 5

Implementation

Contents

5.1 Implementation

Implementation is all about the carrying out, execution, or practice of a plan, a method, or any design for doing something. Implementation is the action that must follow any preliminary thinking in order for something to actually happen. It encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, and running, testing, and making necessary changes.

Here we are going to implement the above algorithms using MATLAB.MATLAB, which stands for **MAT**rix **LAB**oratory[1], is a state-of-the-art mathematical software package, which is used extensively in both academic and industry. It is an interactive program for numerical computation and data visualization, which along with its programming capabilities provides a very useful tool for almost all areas of science and engineering. But unlike

other mathematical packages, such as MAPLE or MATHEMATICA, MATLAB cannot perform symbolic manipulations without the use of additional Toolboxes. It however remains one of the leading software packages for *numerical* computation.

MATLAB® [1] is a high-level language and interactive environment that enables us to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and FORTRAN. We can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions are available separately) extend the MATLAB environment to solve particular classes of problems in these application areas. MATLAB provides a number of features for documenting and sharing your work. We can integrate our MATLAB code with other languages and applications, and distribute our MATLAB algorithms and applications.

Key Features of MATLAB are

- High-level language for technical computing.
- Development environment for managing code, files, and data.
- Interactive tools for iterative exploration, design, and problem solving Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration .
- 2-D and 3-D graphics functions for visualizing data.
- Tools for building custom graphical user interfaces.
- Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, FORTRAN, Java, COM, and Microsoft Excel.

MATLAB provides a high-level language and development tools that let helps us quickly develop and analyze your algorithms and applications [2].The MATLAB language supports the vector and matrix operations that are fundamental to engineering and scientific problems. It enables fast development and execution. With the MATLAB language, we can program and develop algorithms faster than with traditional languages because there is no need to perform low-level administrative tasks, such as declaring variables, specifying data types, and allocating memory. In many cases, MATLAB eliminates the need for 'for' loops. As a result, one line of MATLAB code can often replace several lines of C or C++ code. At the same time, MATLAB provides all the features of a traditional programming language, including arithmetic operators, flow control, data structures, data types, object-oriented programming (OOPs), and debugging features. MATLAB supports the entire data analysis process, acquiring data from external devices and databases, through preprocessing, visualization and numerical analysis, to produce presentation-quality output. The MATLAB product provides interactive tools and command-line functions for data analysis operations [2].

5.2 Implementation of Cryptosystem using Algebraic Geometric Code

Implementation process [4] include algorithms for

a. Parameter Generation
b. Key generation
c. Encryption
d. Decryption
e. Decoding

a. Parameter generation

Parameter generation is an important factor in developing an algorithm based on Elliptic curve and Algebraic Geometric code. Parameters of an elliptic curve is called domain parameters which include field size, a, b of a curve E: $y^2 = x^3 + ax + b$ [3]. Two types of finite field are used in a cryptographic application are Prime field and binary field. In this paper, elliptic curve over prime field F_P is considered and domain parameters of curve include (P, a, b, B, N_B ,h) where P is the field size, a and b are parameters in the equation of curve, B is the base point of the curve, N_B is the order of the base point and h is an integer which is cofactor $h = \#E(F_p)/n$ [5][6][7]

Parameter generation can be done by random method or Kobliz random selection method. When we select a finite field, number of elements on the field should be a large prime [8, 9]. This is to avoid attacks and to improve the security of the system.

```
Function[p. a, b] =Domain parameters(a, b)
1.p ← 1
2. if 4 a^3 + 27 b^2 = 0
    exit
   else
3.p← ECC_prime(a, b)
4.N ←point _count(p ,a ,b)
5. if (is_ prime(N) )
        return(p)
  else
   go to step1
5.end
```

Fig 5.1 Domain parameters

b. Key generation algorithm

This algorithm takes the domain parameters, computes the base point and generates keys. Procedure is as follows

```
function[β]=Genkey (p,a,b)
1.[xB,yB]=Genbasept(a,b,p);
2.m=Findorder(xb,yb];
3. β =randint(1,1,m);
4.End
```

Fig 5.2: Key generation

c. Encryption procedure

Message M is divided into smaller unit's m_1, m_2....m_k and converted into vector or linear set of integers. Encryption procedure includes creation of generator matrix and converting it into an algebraic geometric code. The code is multiplied by the key parameters. The procedure can be described as follows.

```
Function[CM,P] = Encryption(p, a, b, message, β)
1.[xp, yp] = Genbasept(a,b,p);
2.[X,Y]=points(a,b,p)
3 .[U]1xk= msg2int(message);
4.[GM] kxn = Genmatrix(X,Y,a,b,p);
5. s = findorder(xb,yb);
6. α = randint(1,1,s);
7.[CT] = [U] * [GM];
8. γ = α * β;
9. [xk, yk]= Succdob(xb,yb, γ,a,p);
10.K =[xk,yk]
11. P = [x2, y2] = Succdob(xb,yb, α,a,p)
12. CT=[ K *[CM], P];
13.end
```

Fig 5.3 Encryption

The algorithm is an overview of the encryption process using MATLAB. Process involves functions for base point generation, creation of generator matrix and elliptic curve scalar multiplication. Scalar multiplication is done by successive doubling process.

c. Decryption procedure

Decryption procedure involves accepting the cipher text and converting into original message .It involves taking cipher text and decrypt it using private key and publicly available p, a, b.

```
Function[Message]=Decryption(CT, P)
1.Compute k  = Succdob(x2,y2, β,a,p))
2. [X,Y]=points(a,b,p)
3.[GM] kxn =Genmatrix(X,Y,a,b,p)
4.CM = CT/ k;
5.[U] = CM *  pinv[GM];
6.[Message]=initomsg(U);
7.End
```

Fig 5.4. Decryption

The procedure given above contains functions of MATLAB. The program was executed and result is as follows. The example here takes an input field and other random parameters randomly. The Δ here is 31.

Encryption	**Decryption**
INPUT: p = 31, a=1, b=1	INPUT: p = 31, a=1, b=1
(x_b,y_b)= (9,10)	(x_b,y_b)= (9,10)

N = 34	N = 34
r = 31	r = 31
Plain text: ecccryptography CipherText: **79520 24836 45192 58772 72324 78680 77140 73752 23856 80248 5871 81984 60984 78148, 19, 28**	Cipher text: **79520 24836 45192 58772 72324 78680 77140 73752 23856 80248 5871 81984 60984 78148, 19, 28** Decrypted text: ecccryptography

Fig 5.5 A Simple Example

5.3. Performance analysis over various fields

The algorithm is implemented by using Mat lab for various field and executed in an Intel Pentium processor. The system was tested for time required for key generation, encryption E and decryption D. Five fields were randomly chosen and are 13, 31, 83,127 and167. An elliptic curve E is of form $y^2=x^3+ax+b$ and is defined over a finite field F_p and is represented as E_p (a, b).

1.q=13,a=1,b=1 curveE_{13}(1,1) Number of points n=15, base point(12,8) Random key limit: 11 Key generation=0.0630 μs Encryption time=0.008667 μs Decryption time=0.0639 μs
2. q=31,a=1,b=1 CurveE_{31}(1,1) Number of points n =32:,base point(17,31): Random key limit: 31 Key generation =0. 6090 μs Encryption time=0.02 μs Decryption time=0.082 μs

3.q=83,a=1,b=1 $CurveE_{83}(1,1)$ Number of points n =90:,base point(12,9) Random key limit: 89 Key generation =10. 14100 μs Encryption time=0.04264μs Decryption time=0. 55 μs
4.q=127.a=1,b=1 $CurveE_{127}(1,1)$ Number of points n =131, base point(18,3): Random key limit: 131 Key generation =34.1 μs Encryption time=0.06233 μs Decryption time=0.63 μs
5.q=167,a=1,b=1 $CurveE_{167}(1,1)$ Number of points n =147, Base point (35,21) Random key limit: 144 Key generation: 62.8280 μs Encryption time=0.25 μs Decryption time=0.85 μs

Fig 5.6 Time taken over various field.

Performance analysis can be viewed by the following

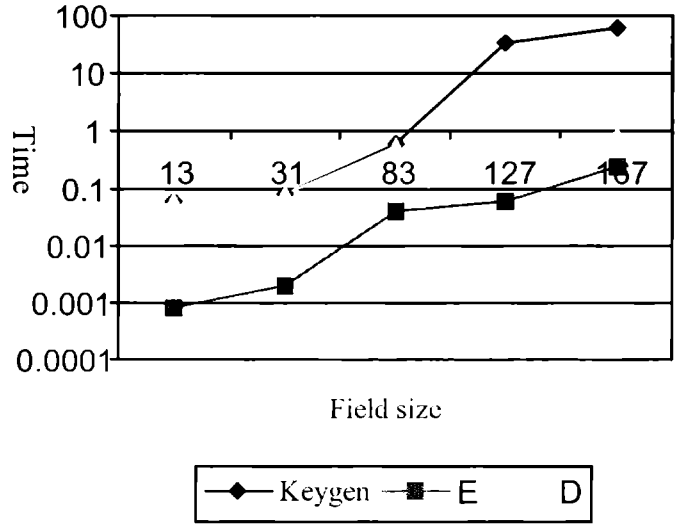


Fig 5.7: Graph showing time requirement for Key generation, Encryption and Decryption

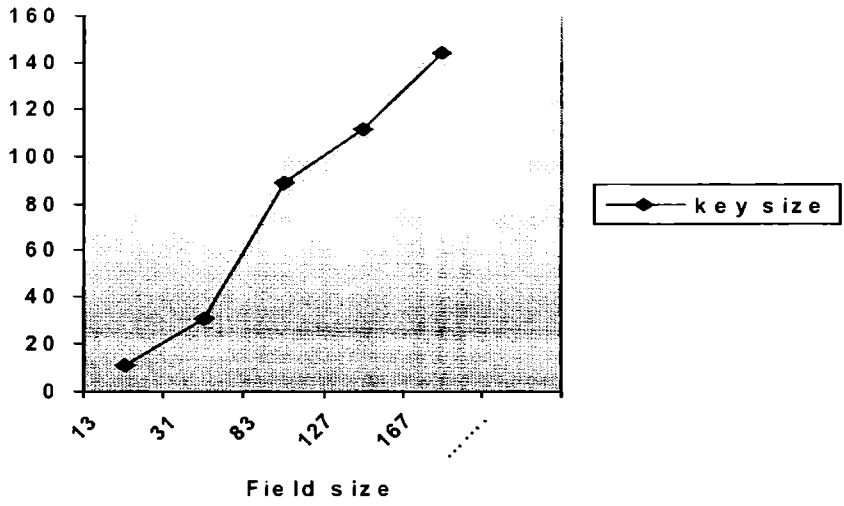


Fig 5.8: Graph showing Field size Vs Key size

From the above graphs we can see that computation time increases as field size increases. Computational time is dependent on factors such as elliptic

curve points, scalar multiplications, and point doubling and generator matrix generation. Above mentioned operations are developed using MATLAB. From these, we can also see that whenever field size increases, security level goes high. So, it can be concluded that, performance and security of a Cryptosystem using Algebraic Geometric code can be improved by selecting a field of sufficiently large prime.

Here we have computed key generation, encryption, decryption and key size for a curve. We can see that computational complexity increases with field size. at the same time security level increases. Whenever we develop a system it should be of higher security. To achieve that, we must have a field of sufficiently large prime. Overhead in computation can be solved by making use of processor of higher capacity.

5.4 Implementation of Cryptosystem using the concepts of repetition codes

Implementation process is done by using MATLAB. Implementation process includes algorithms for

a. Parameter setting
b. Key generation
c. Encryption
d. Decryption
e. Decoding

a. Parameter Setting

Parameter generation is an important factor in developing an algorithm based on Elliptic curve and Algebraic Geometric code. Parameters of an elliptic

curve is called domain parameters which include field size, a, b of a curve E: $y^2 = x^3 + ax + b$. The domain parameters of curve include (P, a, b, B, N_B , h) . P is the field size, a and b are parameters in the equation of curve, B is the base point of the curve, N_b is the order of the base point and h is an integer which is cofactor $h=\#E(F_p)/n$[5,7].

Parameter generation [8, 9] can be done by random method or Kobliz random selection method .When we select a field, number of elements on the field should be a large prime. This is to avoid attacks and to improve the security of the system.

Procedure for generation of curve and its parameters is same as previous algorithm and can be generated using procedure in Fig 5.1.

Here, only 3 main domain parameters are generated. Remaining parameters are generated during the encryption process. Code parameters include (n, k, d), where n is the number of points on the curve, K is the dimension and d is the distance. K is selected according to the linear vector space generated.

b. Key generation algorithm

This algorithm takes the domain parameters, computes the base point and generates keys. Procedure is as follows

```
Function[β]=Genkey (p,a,b)
1.[xB,yB]=Genbasept(a,b,p);
2.m=Findorder(xb,yb];
3. β =randint(1,1,m);
4.End
```

Fig 5.10: Key generation

c. Encryption procedure

Encryption procedure includes creation of generator matrix and conversion of message into points. Here, message is treated as an array of single characters and converted into points. If the size of message is very large, message can be grouped into convenient size and can be converted into points.

```
Function[CM,P]=Encryption(p, a, b, message, β)
1.[xp, yp]=Genbasept(a,b,p);
2 [xp, yp,n]=pcpoints(a,b,p);
3.[xmp, ymp]=msg2points(message);
4.GM=Genmatrix(xmp,ymp,a,b,p);
5. s=findorder(xb,yb);
6. α = randint(1,1,s);
7. γ = α * β;
8. [xp, yp]= Succdob(xb,yb, γ,a,p);
9.P=[xk, yk];
10. [xk, yk]= Succdob(xb,yb, α,a,p);
11. Z=[xk, yk];
12. CM=[GM+ P];
13 Return(CM,Z)
```

Fig 5.11: Encryption

The algorithm is an overview of the encryption process using MATAB. Process involves functions for base point generation, creation of generator matrix and elliptic curve scalar multiplication.

c. Decryption procedure

Decryption procedure involves accepting the cipher text and converting into original message It involves converting output contents into points and then into message.

```
Function[Message]=Decryption(CM, Z,a, b, p)
1.Compute Q=Succdob(X_k ,Y_k, β,a,p)
2.CM'=CM-Q
3.[M_x,M_y]=Cipher2point(CM');
4.len = length(M);
5. for i = 1:1:len
6.        Message[ I ]=point2msg(M_x[i], M_y[j]);
7.  End
```

Fig 5.12 Decryption

e. Decoding Procedure

The algorithm is as follows

```
Function[ ]=Decodc(CM', Z)
1.Compute GM1= Genrmatrixeval(CM');
2.k=size(CM');
4.count=0;
5.for i=1:1:k
6. for j= 1:1:k
        if(GM1( i , j) = = GM1( i+1 , j))
            count++;
         end
7. if ( count > k/2)
      disp('accept ' +GM1( i, j ));
  else
   disp('error in data');
   end
```

Fig 5.13 Decoding

The received message looks like a repetition code, here we are comparing the entries and deciding whether to accept the message or not. Here, we are assuming that the channel error will not effect all the repetitive

information. Various function calls are given in above procedures. These function call contains code to execute corresponding process. The program was executed .The following table shows the output of the above mentioned program for a small field.

Encryption	**Decryption**
INPUT: p = 163, a=1, b=2	INPUT: p = 163, a=1, b=2
(x_b, y_b)=(25,2)	(x_b, y_b)=(25,2)
N_B=87 ; N = 177 , h = 2	N_B=87 ; N = 177 , h = 2
r = 42	r = 42
Plain text: welcome CipherText: 261 113 244 274 251 244 113 241 273 115 112 272 272 273 145 113 258 113 199 258 113 262 273 182 112 184 205 273 127 116 121 112 121 121 116	Cipher text: 261 113 244 274 251 244 113 241 273 115 112 272 272 273 145 113 258 113 199 258 113 262 273 182 112 184 205 273 127 116 121 112 121 121 116 Decrypted text: Welcome

Fig. 5.14: Encryption and Decryption process

Analysis of algorithm was done over various fields on Pentium IV processor. Time taken over various fields during the process of encryption is as follows

Field F_q	127	163	192	223	321	521
Time(s) Encryption	1.16	1.46	3.14	5.44	7.18	9.12
Time(s) Decryption	2.12	2	4.32	4.523	9.12	12.186

Table 5.1: Field size Vs Time for encryption and decryption

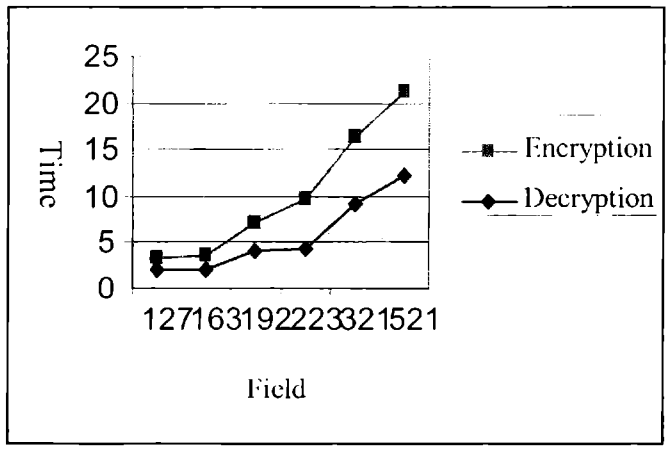


Fig 5.15 Encryption, Decryption time Vs Field

From Fig 5.15, we can see computing time increases with field size. Size of key also increases with field size. Here the disadvantage is the size of the cipher text. This can be overcome by the advantage of decoding process, which helps detecting errors. An analysis has been done for the above algorithm for various message lengths. The result of the analysis can be represented in Fig 5.16 as follows.

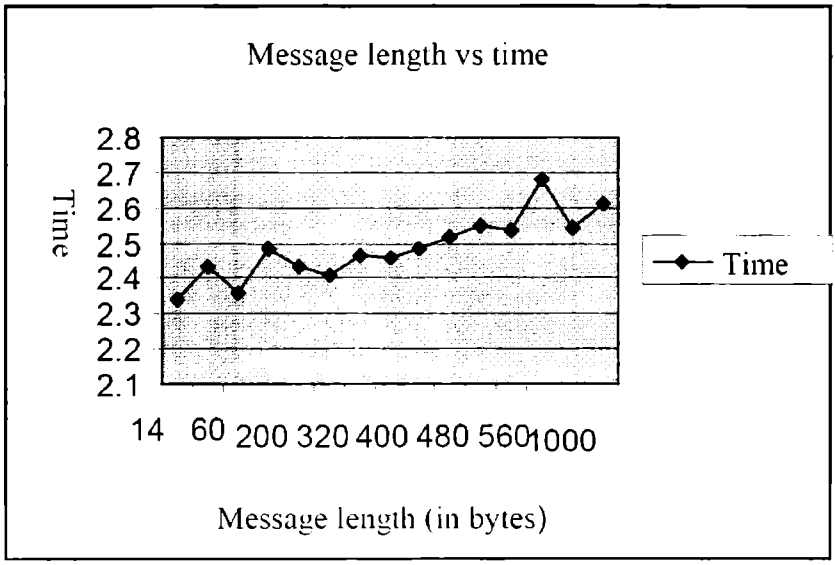


Fig 5.16 Message length vs time

The result in Fig 5.16 shows that, increase in message size does not affect the computation time. The program was executed for a field F_{163} over various message sizes from 14 bytes to 1000 bytes and result show that, there is not much variation in time, when the message size is increased.

The system was tested for the size of the output message. As indicated earlier, outputs represent a repetition code here. So, as the size of the message increases, the output size also increases. The graph below shows length of cipher text vs message size in the above mentioned system.

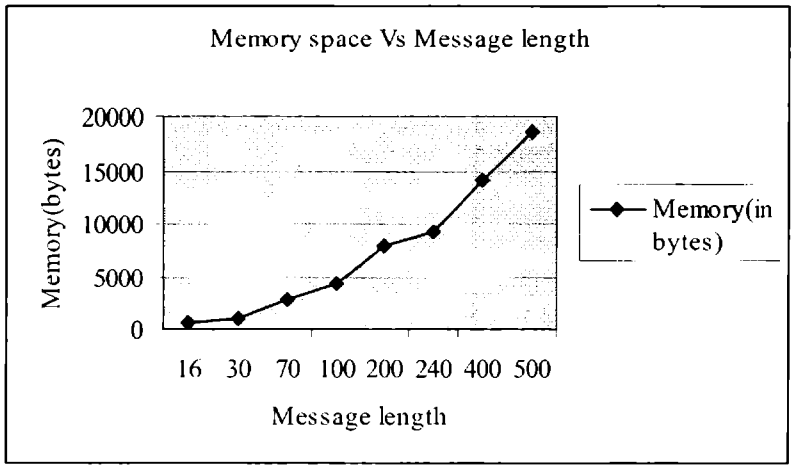


Fig 5.17 Message time vs Message length

From the Fig 5.17 we can see that the memory requirement increases as the message size increases.

5.5 Implementation of the secret sharing method

In this section implementation of the above discussed algorithm is discussed. Implementation was done on MATLAB and various steps are as follows.

a .Secret splitting

```
Function [s]=ssplit(a,b,p,n)
1.[xb,yb]=Genebasepoint(a,b,p);
2.z=findorder(xa,yb,p);
3.[s]=secretplit(randint(1,1,z),m);
End
```

Fig 5.18 secret splitting

The above procedure takes a random integer and splits it into n units as per dealer's requirements. Next procedure shows how the secret is distributed among n users.

b. Secret distribution

```
Function[f]= distribute[s,m, Fn]
1.[Fn]=Generatepoly[s];
2.for i = 1 to m
3.   Compute f( i ) = Fn ( i );
4. end
 End
```

Fig 5.19 Secret Distributing

Step 3 here computes shares of the i th user and the dealer distribute F[i] to the i th user. F_n is a polynomial generated with the help of the splitted shares.

c. Secret reconstruction

As specified earlier, this step is done at the receiving end. The process is as follows. Dealers do this with the shares of information he got from his authorized users.

```
Function[S]=Secretconstruct(A,Fn)
/* f is an array of shares a1....am*/
1.S=0;
2.for i : 1 to m
3.   sj= fi * F-1
4.end
5.for  j=1 to m
6.        S = S + si;
7..end
8.End
```

Fig 5.20 Secret reconstruction

The program was executed and result is as follows. The Function defined here

are the coding function in MATLAB .The security can be assured only when field size is large prime.

Input : a=1, b= 1, n =6
Field p = 167.
Number of points n =147,
Base point (35,21)
Random key limit: 144
Secret key generated is 101
Secret shares to 5 users :f_1=128, f_2= 121 ,f_3= 142 ,f_4=35, f_5=141

Fig 5.21 An Example

5.6 Analysis of various curves in Cryptography

Cryptography can be defined as mathematical techniques related to the process of information security. In 1985 Koblitz and Miller [10, 11] introduced concepts of curves in cryptography. They made use of elliptic curves and the system is known as Elliptic Curve Cryptography. In this chapter we will have a study of elliptic curve, hyper elliptic curve, super singular elliptic curve, Klien quartic curve. We will see how and which of these curves are suitable for the use in cryptography. Cryptography, as specified earlier is the science of security which involves mathematical techniques for the process of encryption and decryption. Koblitz and Miller [10, 11] introduced the concept of the usage of curves in Cryptography. They used the concept of elliptic curves in their cryptography and is known as Elliptic Curve Cryptography. Here we will see various curves that can be used in cryptography and properties of it.

a. Elliptic curves

An elliptic curve E is a curve defined by a nonsingular Weierstrass equation [3]

$$E: y^2+a_1xy+a_3xy = x^3 + a_2x^2+a_4x + a_6$$

where the following equations $a_1y=3x^2 +2a_2x+a_4$; $2y+a_1x+a_3=0$ cannot be satisfied simultaneously by any point (x,y) on the curve E. Elliptic curve that

are used for the purpose of the cryptography are of the form $y^2=x^3+ax+b$. An Elliptic curve is an abelian group with an identity O. Certain properties of elliptic curve are

1. Let P be a point at infinity O, then define –P to be infinity. For any point Q define O+Q to be Q. This serves as an additive identity of group $E(F_q)$.
2. The negative –P is a point with same x coordinate and different y coordinate i.e.

   if P(x, y) then –P is (x,-y)

   if Q= - P then P + Q = O
3. If P and Q are different coordinates then the line l = PQ intersects the curve exactly at one or more point R.
4. If P = Q , Let l be the tangent line to the curve at P and let R the only point that intersect line l with the curve then 2P = -R

From the above said properties we can say that the points of elliptic curves form an abelian group. Various methods are there for generation of curve for use in Cryptography. They include complex multiplication, Kobltiz method and random selection method.

Security [12] of ECC is dependent on Elliptic Curve Discrete Logarithm problem. DLP in ECC can be defined as follows. Given a point $P \varepsilon E (F_q)$ and point $Q \varepsilon E (F_q)$ such that Q = lP. We have to find l. Various attacks are known till date include Pohig-Hellman, Index – Calculus method , Pollard-Rho method[3] .We can make our algorithm secure from all attacks by choosing suitable parameters that can be used in the process of encryption and decryption. Most of the attacks rely on the size of the group [Section 4.6].

Advantages of cryptography using elliptic curve include smaller key size, attacks are less effective, reduced bandwidth, greater efficiency and reduced or simple mathematical operations

b. Hyper elliptic curves

The curve of the form $y^2+h(x)\ y = f(x)$ [13] is called Hyper elliptic curve. Hyper elliptic curve are curves of genus g greater than one where as elliptic curves are curves of genus 1. There exists a hyper elliptic curve whose genus varies from 2 to infinity. For Hyper elliptic curves there are no natural group law on the points on the curve as in the case of elliptic curve by which one can do operations like point addition, scalar multiplication, point subtraction etc. This is because of the fact that, points never from group. But in order to use in cryptography we should make use of some arithmetic properties of curve. Hence for hyper elliptic curve a group law is defined via the Jacobian variety of the curve over a finite field which forms abelian divisors over the divisors group.

The Jacobian of a Hyper elliptic curve [14] C is the quotient group $j = D^0/P$ where D^0 is the set of divisors of degree zero and D is the set of divisors of rational functions. Jacobian of genus g hyper elliptic curve will have q^g points. Koblitiz [13] proposed Picard group $pic^0(c/p)$ of a Hyper elliptic curve as a further group that is suitable for Cryptographic applications. For genus ≤ 4, these curves are secure provided that group order is sufficiently large and that one should avoid curve for which special attacks that are known. Forbenius Automorphism [14] in Hyper elliptic curve can be used to obtain fast arithmetic (especially scalar multiplication). As specified earlier, points of curve doesn't form a group but it can be achieved by Forbenius endomorphism which operates on divisor class in Mumford representation [14] by raising coefficients of polynomial a and b to the q^{th} power.

If points are represented via normal basis over F_q, then computing q^{th} power of a finite element just means a cyclic shift of representation. This computation is performed by almost 2g cyclic shifts. In hyper elliptic curve normal basis representation is more efficient than polynomial basis. In Koblitiz

curve, Picard group of over Fq^n comes along with an automorphism group of order n and inversion and this can be used in crypt algorithms.

The DLP of Jac (F) (Jacobian of F) stated as follows. Give two divisors D1 and D2 ε Jac (F) determine a smallest m such that D2 = mD1. Operation involved in it is group addition and group doubling. Pollard-Rho method and its variants [15] are important examples of algorithms for solving the DLP in generic group with complexity better than O(n) in groups of order n .Operations involved in ECC and HECC are entirely different.

c. Super singular curves

A super singular curve should satisfy following conditions Let $q=p^n$ and let E be an elliptic curve over F_q. Suppose the characteristic polynomial of Forbenius endomorphism $P(x) = x^2 - tx + q$ so that #E (Fq) [number of points] =q+1-t [16] then

i. The endomorphism ring of E (over algebraic closure of F (q) is non-commutative.
ii. E has no points of order p i.e. (E(F(q))={o}.

If C is a genus 2 curve over F_2 the form $y^2+y=f(x)$ where f(x) is monic polynomial of degree 5 then C is a super singular. The general format $y^n = f(x)$. Security in super singular curve is dependent on DLP of divisor group. There is sub exponential algorithm for solving DLP of Super singular curves. Frey-Ruck [15] described how the Tate pairing can be used to map the discrete logarithm problem on the divisor class group of a curve C over a finite field F_q into multiplicative group F_q^k of extension of base field. Menzes, Okamoto and Vanstone [17] showed that the value of divisor class k is always ≤ 6. Because of this attack can be easily done in the cryptosystem. These results show that we should avoid it in cryptography. Selecting certain super singular curve with

certain properties can avoid attacks. But selection itself is difficult. The super singular curves that can be used in cryptography include[17]

G=3 C: $y^3 = x^4 + x^3 + ax^2 + x + \alpha$ over F_2^2 $P(x) = x^6 + 3x^4 + 4x^3 + 12x^2 + 2^6$

G=4 C: $y3 = x^5 + \alpha$ over F_2^2 $P(x) = x^8 - 2x^4 + 16$

d. Klien quartic curve

Klien quartic curve is a homogenous curve X: $ax^3y + by^3x + cz^3x$ which can be considered over any field. This curve is a non-singular curve and genus is 3. Here also Jacobian Jac(x) of curve x is a three dimensional abelian variety, defined over a field $k = F_q$. If coefficients and q are properly chosen, the number of points of group Jac $_k(x)$ is prime which form cyclic group [18]. This property made it possible for cryptographic applications.

Security is similar to divisor DLP. Selection of curve is dependent on certain parameters, so that its points form a cyclic group. It is difficult to find out such a curve.

Here we have discussed about various curves and how these curves can be used in cryptography. Each and every curve has its own advantages and disadvantages while using it in cryptography. One main advantage that is common to all cryptosystem using curves compared to public key cryptosystem is smaller key size. Among these cryptosystem elliptic curve is more secure and has less key size, less computational overhead and less processing power. Apart from these, selection of curve is also very important in Cryptography which in turn depends on parameters. The security of cryptosystem using curves is purely dependent on curve and field. Overall comparison is as shown in the table below.

	Elliptic curve	**Hyperelliptic curve**	**Super Singular curve**	**Klien quartic curve**
1	Smooth projective curve - F_q	Smooth projective curve over F_2^m	Smooth projective curve over F_2^m	Non-singular curve defined over any field.
2	Points form an abelian group	Jacobian forms an abelian group.	Jacobean forms an abelian group.	Jacobian forms an abelian group.
3	Genus g=1	g≥1	g≥1	g=3
4	Polynomial basis	Normal basis	Normal basis	Polynomial or normal basis
5	Easy to find curve suitable for cryptography	Only certain curves are suitable for cryptography	Only certain curves are suitable for cryptography	Few points forms a cyclic group that curves can be used in cryptography
6	Cryptographic operation depend on points on curve	Cryptographic operations depend on divisors of the points on curve	Cryptographic operations depend on divisors of the points on curve	Cryptographic operations depend on divisors of the points on curve. But curve is suitable for generation of suitable codes in algebraic geometry.
7	Security depends on dlp on points	Security depends on dlp on divisors	Security depends on dlp on divisors	Security depends on dlp on divisors
8	Cryptographic algorithms include key generation, encryption and decryption	Cryptographic algorithms include divisor generation, key generation, encryption and decryption	Cryptographic algorithms include divisor generation, key generation, encryption and decryption	Cryptographic algorithms include divisor generation, key generation, encryption and decryption

Table 1.2 Comparison of various factors of curve

5.7 References

[1] http://www.math.iitb.ac.in/~pde08/matlab1.pdf

[2] http://www.mathworks.com

[3] Hankerson, Menzes, Vanstome, "Guide to Elliptic Curve Cryptography",Springer, CRC press(2004).

[4] G.B Agnew R.C. Mullen and A Vanstome, "An implementation of elliptic curve cryptosystem over F_2^{55}". IEEE journal on Selected Areas in Communications, Vol. 11, No.5, 1993.

[5] Certicom, " An Introduction to Information security", The first in series of ECC white paper, Mareh 1997.

[6] W.Diffie, M.E Hellmann, "New directions in Cryptography", IEEE transactions for Information theory,Vol.22, pp.644-654,1976.

[7] Menezes,A.J Van Oorschot ,P.Vanstome, "Handbook of applied Cryptography", CRC press-1997.

[8] A.Lenstra and E.Verheul, "Selecting Cryptographic Key sizes", Proceedings of PKC 2000, LNCS 1751,pp 445-466 Springer-Verlag 2000.

[9] "Standard specifications for public key Cryptography", IEEE p1363/D8 [draft version 8].

[10] V. Miller, "Uses of elliptic curve in cryptology", Proceedings of crypto'85, LNCS 218, pp 417-426, New York: Springer-Verlag 1986.

[11] N.Koblitz, "Elliptic curve crypto systems", Mathematics of Computation, 48, pp. 203-209.1987.

[12] "Remarks on security of ECC", a Certicom white paper published Sep-1997.

[13] N.Kolbliz, "Hyper Elliptic cryptosystem", Journal of Cryptography ,1989 pp.139-150.

[14] R.M Avanzi, "Aspects of Hyper elliptic curves over a large prime field in software implementation of Cryptography", e-print archive report 2003/253, 2003 .

[15] G.Frey and H.G Ruck , " A remark concerning m-divisibility and discrete logarithm in divisor class of group of curves", Mathematics of Computation 62, 2006.

[16] J.H Silvermann, "Arithmetic of elliptic curves", Springer GTM 106,1986.

[17] A.J Menzes, T.Okamoto and S.A Vanstome, " Reducing elliptic curve algorithm to logarithms in finite field", IEEE transactions on Information Theory, 39, No.5, pp.1639 – 1646,1993.

[18] G.Lachaud, "The Klein quatric as a cyclic group gene", Mosc. Math Journal, Vol .5(4), pp.857-858, 2005.

.....

Chapter 6

Concluding Remarks and Some Outlooks

Contents

6.1 Comparative Analysis

In this thesis we have seen various public key cryptosystems, cryptosystems using codes and cryptosystem based on algebraic geometric code using elliptic curves. From this we can see that most of public key Cryptosystem including Elliptic curve cryptography are secure. The cryptosystem based on the concepts of coding theory makes use generator matrix as key, resulting in increased key size. Various studies done in previous chapters, shows that to be secure field size should be a large prime. As field size increases, sizes of generator matrix also increase. This shows that existing cryptosystem using Algebraic Geometric code cannot be effectively used because of above mentioned problems. Again in the existing system, encryption and decryption are done as part of encoding and decoding.

In the Cryptosystem developed here the concepts of key is based on parameters of elliptic curve. Because of this the key size is very small compared to other existing systems based on Algebraic Geometric code. The process of encryption, encoding, decoding and decryption is done separately. By doing this, we can ensure more security to the information transmitted.

The table below shows a comparative analysis of ECC, existing cryptosystem using algebraic geometric code and the proposed cryptosystems. The security

of the elliptic curve cryptosystems and the proposed systems are based on discrete logarithm problem. So field size should be of at least 1024 bits in order to have the DLP unfeasible to solve. The groups of points on elliptic curve are also chosen in such a way that attacks likes index-calculus. Pollard –rho methods are unfeasible to solve. The security level of the cryptosystem is based on the number of operations required to resolve the problem in a reasonable amount of time. That is if we take a field of 1024 the security level then security level will be approximately 2^{80}.

Parameters	**ECC**	**Mc-Eliece**	**Cryptosystem using AGC**	**Cryptosystems Using repetition codes**
F_q : 1024				
Key size(bits)	163	1024 x 512	163	163
Err correction Capability	-	50	280	250
Security level(bits)	>80	56	>80	>80

Table: 6.1 Comparative studies over F_q: 1024

Parameters	**ECC**	**Mc-Eliece**	**Cryptosystem using AGC**	**Cryptosystems Using repetition codes**
Fq : 2048				
Key size(bits)	224	2048 x 1024	224	224
Err correction Capability	-	100	≅500	≅500
Security level(bits)	>112	80	>112	>112

Table: 6.2 Comparative studies over F_q : 2048

From the above table we can see that key size, error correction capability for code generated using elliptic curve is far better than the code generated using other curves and other algebraic geometric concepts. The execution time will be high compared to ECC. The error detecting capability

for the system developed using the concept of repetition code is very high compared to other systems but bandwidth is comparatively high here. The security of the system is inevitable and can be obtained by selecting curve and field as mentioned in Chapter 4. The table below shows a comparative analysis of cryptography and cryptosystem using Algebraic Geometric codes (AGC).

Parameters	**Public Key Cryptography**	**AGC in Cryptography**
1.Security	Secured	Secured
2.Reliability	Not reliable	Reliable
3.Error correction and detection	Not possible	Possible
4.Key size	Depends on the chosen Cryptographic methods	Small key size
5.Amount of Information transmitted	Less amount of information	Comparatively large amount of information
6.Computation	Less	High
7.Attacks	Prone to attacks*	Less prone to attacks
8.Security	Depends on discrete logarithm problem	Depends on Elliptic Curve Discrete logarithm problem bust also on structure, function field and finite field

Table: 6.3-Comparison- Public Key cryptography with AGC in cryptography

*A public key cryptosystems (Asymmetric crypto systems) uses two keys : public key, which is known to the public, and private key which is known only to the user. User A uses the received public key to encrypt its messages. Then when the encrypted messages arrive, User B uses its private key t.o decrypt them. The method followed above reduces the problem of symmetric key cryptosystems in managing single key, but this unique feature of public key encryption makes it mathematically more prone to attacks. This is because the secret key generation procedure mainly follows integer factorization problem or discrete logarithm problem. The intruder may try to attack the system to get the private key with help of the public key

6.2 Conclusion

Communication is the process of transmitting information from one place to another through a channel. The channel/medium can be cable, satellite, wireless networks or optical fibers, which are highly susceptible to noise and other disturbances, compromising the quality of information being transmitted. Apart from this, there are other prominent factors which intentionally make the channel unsecure for critical communication. Main aim of this thesis is to generate a cryptosystem that can ensure not only information security but also error detection and subsequent correction. This is possible with the help of algebraic geometric code. "Algebraic Geometric Code and their relation to Cryptography using Elliptic curves" generate two cryptosystems based on algebraic geometric codes. These algorithms not only introduce security but also preserve reliability of information being transmitted.

A digital signature based on the concepts has been developed so as to prove the authenticity of data received. The sharing of secret is also done so that secret can be derived from a set of authorized users.

So we can conclude that secrecy and reliability of the information can be achieved, with a smaller key size, at a lower cost.

6.3 Future Prospects

In previous chapters, the characteristics of the proposed algorithms have been studied and compared using MATLAB simulation. But this approach uses a general purpose processor programmed to work as a crypto system compromising on overall efficiency. So in order to offload processing complexity and to make the encryption/decryption process faster, an Elliptic Curve Processor which uses hardwired multiplier & adder for handling elliptic curve operations can be used (Field Programmable Gate Array - FPGA/Application Specific Integrated Circuit-ASIC). This offloads computational complexity from the processor to the proposed ASIC, which will speed up overall performance, as it uses a specially made processor capable of

doing operations on elliptic functions effectively. The security of the system is highly dependent on field size. As we increase the field size security can be tremendously increased. But when we increase the field size, computational complexity will also be increased. Here by offloading the computational complexity to an elliptic curve processor, improved overall efficiency and high level of information security can be achieved.

.....ꕥ.....

List of Publications

The following papers were published / presented as a part of the subject

1. "Performance analysis of cryptosystems using algebraic geometric code over various fields Fp." – Published in International journal of Computer Science and Network Security –ISSN 1738-7906 November 2009.

2. "A Cryptosystem using the Concepts of Algebraic Geometric code" - published in Journal of Computer Science, Science publications, ISSN -1549-3636, Vol (6), No(3), March 2010.

3. "Security issues on Cryptosystem based on Algebraic geometric code using Elliptic Curves"- Conference proceedings IEEE-CSI National Conference on Information and Software Engineering 2010 , Aarupadai Veedu Institute Of Technology, Chennai, February 26 – 27 2010, ISBN: 978-93-80043-80-7.

4. "A Digital Signature system based on Algebraic Geometric code using Elliptic curves "(Communicated)

Appendix - A

1. Here are some graphs of curve developed using the MATLAB. The curve is defined over a finite field Fq and of the form $y^2 = x^3 + ax + b$ and is represented as $E_q(a,b)$.

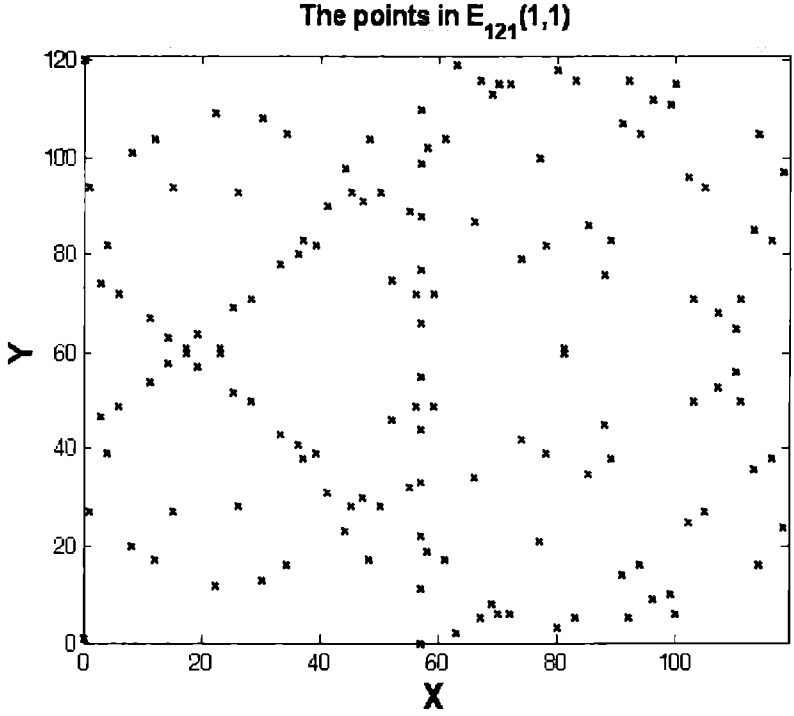


Fig.1 Number of points-144

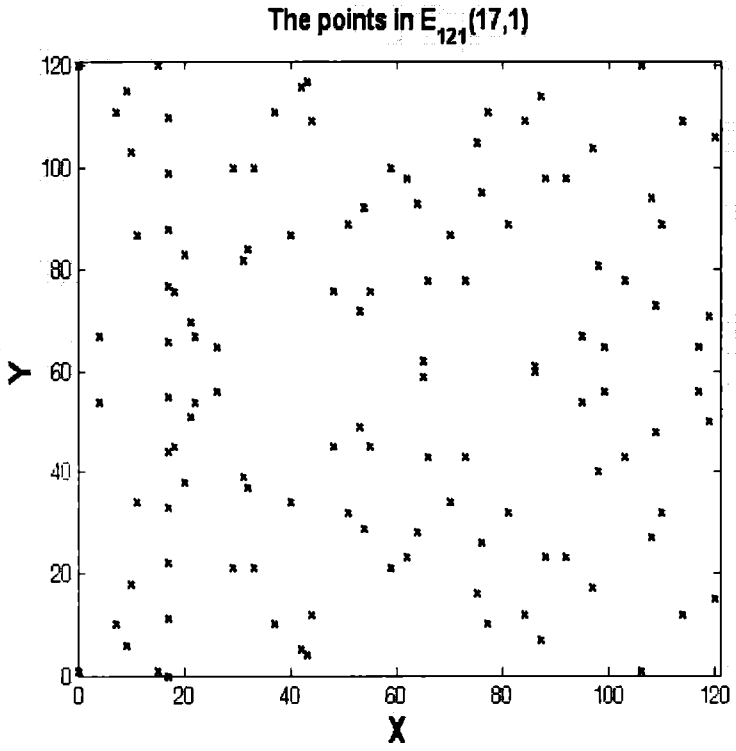


Fig 2 Number of points -122

From the below mentioned figures we can see that number of points of the curve is not only dependent on the size of the field but also on the parameters a and b. size of the field but also on the parameters a and b.

II. 1.Here is some of the sample MATLAB code for generation of curves and certain operations done using MAT Lab in this thesis

/* This is a MATLAB function for generation of curve along with points. The p is the field size, a and b are the parameters of the curve $y^2 = x^3 + ax + b$. The function generates the curve generates the points along with number of points on curve. It also plot the points on graph. */

```
function [points,n] =  curvepoints (a, b, p)
points=zeros(1,1);
```

```
R1= zeros(3,1);
L1 = zeros(3,1);
X = zeros(2,1);
Y = zeros(2,1);
for i=0:1:(p-1)
R1(i+1) = (i)^3 + a*(i) + b;
R1(i+1) = rem(R1(i+1),p);
L1(i+1) = (i)^2;
L1(i+1) = rem(L1(i+1),p);
end
ii=1;
for z=0:1:(p-1)
I=find(R1==z);
J=find(L1==z);
e1 = isempty(I);
e2 = isempty(J);
if (e1) == 0
if( e2) == 0
n=length(I);
m=length(J);
for h=1:1:n
for g=1:m
X(ii)=I(h)-1;
Y(ii)=J(g)-1;
P=[X(ii),Y(ii)];
points(ii)=P;
ii=ii+1;
end, end ,end end end
n=length(X) + 1;
disp(n);
%Generation of curve as points on a graph%
M=plot(X,Y,'x');
z = [X,Y];
disp('    X     Y');
disp(z);
set(h(1),'LineWidth',2);
xlabel('X','FontSize',14,'FontWeight','regular');
ylabel('Y','FontSize',14,'FontWeight','regular');
```

II.2 function [x3,y3,m] = ECADD(x1,y1,x2,y2,a,p)
% This function performs Elliptic Curve addition over an elliptic curve $y^2 = x^3 + ax + b$ .Here we are adding two points $P_1(x_1,y_1)$ and $P_2(x_2,y_2)$ to get a third point $P_3(x_3,y_3)$ which is the sum of P_1 and P_2.

```
%
if x1==Inf
x3=x2; y3=y2;
return
end
if x2==Inf
x3=x1; y3=y1;
return
end
if x1==x2
if y1==y2
if y1==0
display('X3 is infinity')
x3=Inf;, y3=Inf;
return
end
m = sym( (3*(x1)^2 + a)/(2*(y1)) );
n = 3*(x1)^2 - a;
d = 2*(y1);
m = mod( (n * inverse(d, p)) , p );%*Slope of the curve is generated *%
 x3 = mod( (m^2 - x1 - x2) , p);
y3 = mod( (m*(x1 - x3) - y1) , p);
return
end
display('P3 is infinity');
x3=Inf;, y3=Inf;
return
end
n = y2 - y1;
d = x2 - x1;
m = mod( (n* inverse(d, p)) , p);
x3 = mod( (m^2 - x1 - x2) , p);
y3 = mod( (m*(x1 - x3) - y1) , p)
```

Appendix –B

I. Hasse's Theorem

Hasse's theorem states that Let $t = q + 1 - \#E(F_q)$. Then

$$\#E(F_q{}^k) = q^k + 1 - \alpha^k - \beta^k$$

where $1 - tx + qx^2 = (1 - \alpha x)(1 - \beta x)$.

II. Chinese remainder Theorem

Theorem 1: For a,m ☐ Z such that ax = 1 mod m if and only if gcd(a,m) = 1.

Proof: There is a,x ☐ Z such that ax ≡ 1 mod m ⇒ there are x,y ☐ Z such that ax-my=1.

Suppose $m_1, \ldots, m_r$ ☐ N are relatively prime in pairs, i.e. gcd $(m_i, m_j) = 1$ for $i \neq j$. Let $a1, \ldots, ar \in Z$. Then, the system of r congruences is given by

$$x \equiv a_i \pmod{m_i} \quad (1 \leq i \leq r)$$

has a unique solution modulo $M = m_1 \times \ldots \times m_r$ given by

$$x = \sum_{i=1}^{r} a_i M_i y_i \bmod M$$

where $M_i = M/m_i$ and $M_i y_i \equiv 1 \bmod m_i$.

Proof: Note that M_i is the product of all m_j where j = i.So if $j \neq i$ then $M_i \equiv 0 \bmod m_j$ .Note also that ged$(M_i, m_i) = 1$, so by Theorem 1, $M_i y_i \equiv 1 \bmod$ mi has a solution y_i. Thus, $x = \sum_{i=1}^{r} a_i M_i y_i \equiv a_i M_i y_i \equiv a_i \bmod m_i$ for all i, $1 \leq i \leq r$. Therefore, x is a solution to the system of congruences.

III Euler's function

Euler's function $\varphi : N \rightarrow N$ is defined as

$$\varphi(m) = \#\{k \in N \mid 1 \leq k \leq m, \ ged(k, m) = 1\}$$

IV Schoof 's Algorithm

In 1985, Schoof presented a deterministic algorithm that could compute #E(Fq) (its precise value; not a bound or an estimate) in O(1og q) bit operations (where Fq is a finite field of characteristic = 2, 3)

1. Let $l_1 = 3, l_2 = 5, l_3 = 7, \ldots, l_k$ be the k consecutive primes starting at 3, where k is the largest integer such that

$$\prod_{i=1}^{k} l_i \leq 4\sqrt{q} \text{ set } L = l_k$$

2. Compute $\tau_i \pmod{l_i}$ for all i ($1 \leq i \leq k$).

3. Use the Chinese Remainder Theorem to compute

$T = \sum_{i=1}^{k} \tau_i M_i y_i \bmod M$ where $M = \prod_{i=1}^{k} l_i$, $M_i = M / l_i$ and $M_i y_i \equiv 1 \bmod l_i$.

Find a t that satisfies $|t| \leq 2$ (Hasses theorem) i.e if $t > 2\sqrt{q}$ set $t = t - M$.

4. Compute $\#E(F_q) = q + 1 - t$.

V Koblitz's random Selection method

1. Randomly select three elements from F_q; Let them be x,y,a.

2. Set the value for b by computing $b = y^2 - (x^3 + ax)$ since curve equation (1.5) is $y^2 = x^3 + ax + b$.

3.Check that cubic on the right side of equation 1.2, so that is should not have multiple roots, i.e. check that $4a^3 + 27b^2 \neq 0$.

If condition in step 3 is not met, return to step 1.

Else set P = (x, y) and let $y^2 = x^3 + ax + b$ be our elliptic curve

V1. Koblitz's construction algorithm

1. Randomly choose a large prime q.
2. Use Koblitz's random selection method to find an elliptic curve $E(F_q)$ of the type defined in (2.4).
3. Use Schoof algorithm to compute $\#E(F_q)$.
4. Verify that $\#E(F_q)$ is a (large) prime.
5. If step 4 is not satisfied return to step 2.

If Koblitz algorithm is performed, then any point in E other than O would be the generator of E and the ECDLP over E will be intractable.