IJMS 2016 vol. 3 (2): 99 - 110

International Journal of Multidisciplinary Studies (IJMS)

Volume 3, Issue 2, 2016

Elliptic Curve Cryptography and Coding Theory

Sanjeewa R and Welihinda BAK

Department of Mathematics, Faculty of Applied Sciences, University of Sri Jayewardenepura, Sri Lanka

ABSTRACT

From the earliest days of history, the requirement for methods of secret communication and protection of information had been present. Cryptography is such an important field of science developed to facilitate secret communication and safeguard information. Cryptography is based on mathematics. It is an application of different disciplines such as Algebra, Number Theory, Graph Theory etc. Private key cryptography and Public key cryptography are the two main types of cryptography. Public key cryptosystems offer more security and convenience for the users. The main objective of this study is to explore the possibilities of further improvement of Elliptic Curve Cryptography (ECC) by studying the mathematical aspects behind the "Elliptic curve cryptosystem" which is one of the latest of this kind and develop a computer program to generate the cyclic subgroup of a given elliptic curve defined over a finite field $\mathbb{Z}_p$, where p is a prime, which is the major requirement to perform ECC and then use the same to illustrate how data security is achieved from this. For an elliptic curve defined over a field, the points on an elliptic curve naturally form an abelian group. Elliptic curve arithmetic can be employed to develop a variety of Elliptic curve cryptographic schemes such as key exchange, encryption, digital signatures and specific construction of a keyed-Hash Message Authentication Code (HMAC) which are illustrated through this study. Moreover this study proposes an improvement for the encryption of a message through utilization of a concept in "Coding Theory" of Abstract algebra which offers an additional shield for the transmitted message.

KEYWORDS: *Abelian group, cyclic subgroup, ECDH, ECDSA, AES*

Corresponding author: B. A. K. Welihinda, Email: kasuniwe@gmail.com

1. INTRODUCTION

Cryptography is an important field of science developed to facilitate secret communication and safeguard information. Private key cryptography and Public key cryptography can be identified as the two major categories of cryptography. In Private key cryptography a single key is used for both encryption and decryption of messages which renders the inconvenience of having to agree on a common key by the communicating parties prior to the communication. Thus in order to overcome this inconvenience Public key cryptosystems were introduced which involves a pair of keys, namely the Private key and the Public key. The concept behind Public key cryptography is as follows.

The two communicating parties, say Alice and Bob, first have to generate each a pair of keys, the Private key and the Public key. The Public key of each of them is made public and the Private keys are kept as a secret. Then if Alice needs to send Bob a message she should encrypt it using Bob's Public key which can only be decrypted using Bob's Private key and vice versa.

Elliptic curve cryptosystem is the latest Public key cryptosystem of attraction for many mathematicians and computer specialists at present. The security offered by Elliptic curve cryptosystems prove to be unbreakable rather than the security offered by RSA cryptosystem (which was introduced prior to the ECC) and moreover it allows the usage of smaller key sizes while offering the same level of security as RSA which can be considered as its main attraction.

This study was conducted with the intension of meeting the following objectives:

- Study of mathematical aspects behind Elliptic Curve Cryptography.
- Deriving a cyclic subgroup, which is the major requirement to perform Elliptic Curve Cryptography from a given elliptic curve on a finite field $\mathbb{Z}_p$, where p is a prime and developing a computer program as well.
- Development of a computer program to illustrate how security is achieved and implemented in Elliptic Curve Cryptography and make possible improvements to the security of Elliptic Curve Cryptography.

2. BACKGROUND

Elliptic curve cryptography is based on elliptic curves defined over a field. An elliptic curve is a curve given by an equation of the form,

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

with the requirement that the discriminent is non-zero.

$$i.e. \quad \Delta = 4a^3 + 27b^2 \neq 0$$

The above form of the equation is known as the *Weierstrass normal form* for elliptic curves. An elliptic curve is non-singular, meaning that its graph has no cusps or self intersections and it is always symmetric about the x-axis.

Elliptic curves can have points with coordinates in any field such as $\mathbb{Z}_p, \mathbb{Q}, \mathbb{R}$ or $\mathbb{C}$ and for the purpose of cryptography the set E(**K**) of the points on an elliptic curve defined over a field **K** together with a point $\boldsymbol{O}$ which is defined to be the point at infinity is considered.

$$\mathrm{E}(\mathbf{K}) = \{(\mathrm{x}, \mathrm{y}): y^2 = x^3 + ax + b\} \cup \{\boldsymbol{O}\}$$

The points on an elliptic curve naturally form an abelian group and the group law can be constructed geometrically. To perform cryptography it is necessary to obtain a cyclic subgroup of this abelian group.

2.1. Geometry of Elliptic Curves

Consider $E(\mathbf{K}) = \{(x, y): y^2 = x^3 + ax + b \ (\text{modp})\} \cup \{\boldsymbol{O}\}$, the set of points on the elliptic curve $y^2 = x^3 + ax + b$ over $\mathbb{Z}_p$.

- Adding distinct points on an elliptic curve:

Let $P \equiv (x_P, y_P)$, $Q \equiv (x_Q, y_Q)$ be two distinct points on the elliptic curve,
$R \equiv (x_R, y_R)$ be the point representing P+Q,
S be the slope of the line L(x) through P and Q.

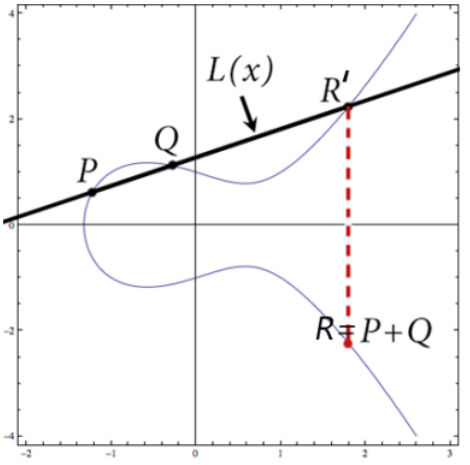

Figure 1. Addition of two distinct points on the elliptic curve

The sum of the two points P and Q on the elliptic curve is defined to be the reflection of the point of intersection (R') of the elliptic curve and the line L(x) through P and Q, on the x-axis. The coordinates of the point R can be obtained using the following formulae given any two distinct points P and Q on the elliptic curve.

$$S = \frac{(y_P - y_Q)}{(x_P - x_Q)} \ (\text{modp})$$
$$x_R = S^2 - (x_P + x_Q) \ \ (\text{modp})$$
$$y_R = S(x_P - x_R) - y_P \ \ (\text{modp})$$

A remarkable property for elliptic curves is that when the elliptic curve is defined over a field **K** with a, b ∈ **K** and if P, Q have coordinates in **K** then P+Q, 2P (or 2Q) also have their coordinates in **K**.

- Adding a point to itself on an elliptic curve:

Let $P \equiv (x_P, y_P)$ be a point on the elliptic curve,
$R \equiv (x_R, y_R)$ be the point representing 2P,
S be the slope of the line L(x) tangent to the curve at P.

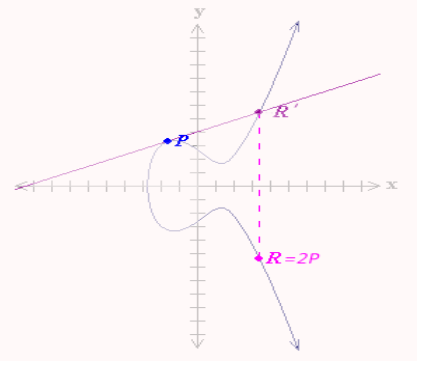

Figure 2. Addition a point to itself

The resultant of the addition of a point to itself (or point doubling) is defined to be the point on the elliptic curve obtained as the reflection of the point of intersection of the tangent line L(x) with the curve, on the x-axis. The coordinates of the point R can be obtained using the following formulae,

$$S = \frac{(3x_P^2 + a)}{2y_P} \ (\text{modp})$$
$$x_R = S^2 - 2x_P \ \ (\text{modp})$$
$$y_R = S(x_P - x_R) - y_P \ \ (\text{modp})$$

- Vertical lines and the extra "point at infinity":

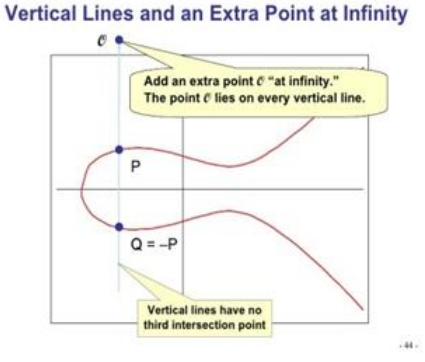

Figure 3. Point at infinity

A vertical line through any point P on the elliptic curve intersects the curve at another point, say Q. Then Q is defined to be –P.

In the view of the above defined geometry for the addition of two points on an elliptic curve we expect the line through the points P and –P to intersect the elliptic curve at another point enabling to obtain the value P+(-P) but it doesn't make such an intersection.

Thus an extra point is created; $\boldsymbol{O}$ **"the point at infinity"** and is defined to be the sum P + (-P).

Important rule: $\boldsymbol{O}$ is a point on every vertical line.

2.2. Algebra of Elliptic Curves

Following the definitions for geometry of elliptic curves the addition law on the set,

$$\mathrm{E}(\mathbf{K}) = \{(\mathrm{x},\mathrm{y}): y^2 = x^3 + ax + b \pmod{p}\} \cup \{\boldsymbol{O}\}$$

satisfy the following properties,

a) $\mathrm{P} + \boldsymbol{O} = \boldsymbol{O} + \mathrm{P} = \mathrm{P}$ $\quad \forall \mathrm{P} \in \mathrm{E}(\mathbf{K})$

b) $\mathrm{P} + (-\mathrm{P}) = \boldsymbol{O}$ $\quad \forall \mathrm{P} \in \mathrm{E}(\mathbf{K})$

c) $\mathrm{P} + (\mathrm{Q} + \mathrm{R}) = (\mathrm{P} + \mathrm{Q}) + \mathrm{R}$ $\quad \forall \mathrm{P}, \mathrm{Q}, \mathrm{R} \in \mathrm{E}(\mathbf{K})$

d) $\mathrm{P} + \mathrm{Q} = \mathrm{Q} + \mathrm{P}$ $\quad \forall \mathrm{P}, \mathrm{Q} \in \mathrm{E}(\mathbf{K})$

making (E(**K**),+) in to an abelian group.

All the Public-key cryptosystems are based on some mathematical problem that is easy to solve but hard to reverse. Likewise Elliptic curve cryptosystem is also based on a mathematical problem which is difficult to be solved in the reverse direction, known as "Elliptic Curve Discrete Logarithm Problem (ECDLP)".

2.3. Discrete Logarithm Problem (DLP)

Let G be a group and $g \in G$. The Discrete Logarithm Problem for G is,

Finding an integer m, given an element h in the subgroup generated by g, which satisfies, $h = g^m$.

The DLP is considered to be computationally intractable. Generally, no efficient algorithm exists which can compute the Discrete Logarithms. But there are certain groups for which DLP can be solved very easily. $\mathbb{Z}/m\mathbb{Z}$ under addition and $\mathbb{R}$ or $\mathbb{C}$ under multiplication are some such groups.

For the group $\mathbb{Z}_p$ under multiplication, solving DLP is considered to be very difficult. The best known algorithm to solve DLP in $\mathbb{Z}_p$ under multiplication takes time $O(e^{c\sqrt[3]{(\log p)(\log\log p)^2}})$. This is called subexponential, since it is faster than exponential (in $\log p$) and slower than polynomial. For cryptographic purposes it is recommended to use a group G for which solving DLP takes time exponential in the order of G.

The Discrete Logarithm problem for points on an elliptic curve is the ECDLP.

2.4. Selection of a suitable generator point to generate a cyclic subgroup

For the implementation of Elliptic curve cryptosystem, it is necessary to generate a cyclic subgroup of the group of points on the elliptic curve. Higher the order of the cyclic subgroup generated, higher will be the security offered by the system. To choose a suitable generator point to generate the cyclic subgroup, it is appropriate to utilize the following algorithm.

1. Calculate the order *N* of the elliptic curve.
2. Out of the divisors of *N*, choose the largest prime divisor as *n*. Then *n* will be the order of the cyclic subgroup to be generated.
3. Compute $h = \frac{N}{n}$.

(*According to the Lagrange's theorem, h is always an integer. h is known as the* ***Cofactor*** *of the subgroup*).

4. Select a random point P on the elliptic curve.
5. Compute $G = hP$.
6. If $G = \boldsymbol{O}$, go back to step 4. Otherwise, G is the suitable generator point of the cyclic subgroup.

The above algorithm works only if n is a prime, since if n was not a prime, the order of G could be one of the divisors of n.

The following parameters are the specific domain parameters required to function any algorithm under ECC.

- The **prime *p*** that specifies the size of the finite field.
- The **coefficients *a* and *b*** of the elliptic curve equation.
- The **base point *G*** which generates the cyclic subgroup.
- The **order *n*** of the subgroup.
- The **cofactor *h*** of the subgroup.

In the Elliptic curve cryptosystem,

1. The **private key** is a random integer ***d*** chosen from $\{1, 2, \ldots, n-1\}$.
2. The **public key** is the point $\boldsymbol{H = dG}$.

If d and G are known (along with the other domain parameters), computing H is "**easy**". But if only H and G are known, finding the private key, d is "**hard**" because it requires solving of the ECDLP.

2.5. Elliptic Curve Diffie – Hellman (ECDH) protocol

Under ECC the key exchange required for user authentication can be accomplished through Elliptic Curve Diffie – Hellman key agreement protocol. It is actually a variant of the Diffie - Hellman algorithm for elliptic curves. Through key exchange a shared secret key can be generated for each communicating party and it not only establishes a successful user authentication but also the x-coordinate of the shared secret key can be used for encryption of the message.

The process taking place in ECDH protocol is as follows.

1. First, the two communicating parties, say Alice and Bob, have to generate each a pair of keys, the private key and the public key (Let d_A, $H_A = d_A G$ be the private and public keys of Alice and d_B, $H_B = d_B G$ be the private and public keys of Bob respectively).

Both Alice and Bob are using the same domain parameters: the same base point G on the same elliptic curve defined over the same finite field.

2. Then Alice and Bob should exchange their public keys H_A and H_B (may be over an insecure channel). An interrupter can intercept these keys but he is unable to find the private keys, d_A nor d_B without solving the discrete logarithm problem which is "hard".

3. Alice calculates $S = d_A H_B$ (using her own private key and Bob's public key), and Bob calculates $S = d_B H_A$ (using his own private key and Alice's public key). S is the shared secret and is same for both Alice and Bob.

$$S = d_A H_B = d_A(d_B G) = d_B(d_A G) = d_B H_A = S$$

Generation of the same shared secret by both the parties indicates successful user authentication. Successful authentication then allows a communication process to take place between the respective parties.

The required message to be communicated can be encrypted using a suitable encryption technique. In this study as an attempt to improve the security of an encrypted message a

concept of a **"Double Encryption"** was introduced. There as the initial encryption Advanced Encryption Standards (AES) encryption was performed and as the secondary encryption the result of AES encryption was subjected to an encryption based on the concept of (*m, 3m*) – Repetition code found in Abstract algebra.

2.6. Advanced Encryption Standards (AES)

AES is implemented using binary arithmetic and consists of a precise mathematical description. It executes a certain number of repeated steps or rounds which is dependent upon the size of the initial key. The AES encryption process is designed to use initial key words of lengths 128, 192 or 256 bits stored one byte at a time as the columns in an initial key matrix with four rows. For simplicity the initial key was fixed to be a key of 128 bits (16 letters) in this study.

When using AES, information is transmitted in binary form. Eight binary digits together form a unit called a *byte*, which will represent a single character in this study. For clear implementation using MATLAB R2013a and simplicity, in this study it was assumed that all the messages were written using only the upper case characters of the English alphabet. Under AES each character was associated with its corresponding element in the ring $R = \mathbb{Z}_{26}$ under a bijection defined as $\alpha : L \to R$, where $L = \{A, B, \dots, Z\}$. (i. e $A \mapsto \bar{0}, B \mapsto \bar{1}, \dots, Z \mapsto \overline{25}$). However, in order to use bytes to represent each of these characters the bijection α has to be extended as described in the following example.

As an example when the letter G is considered, it gets mapped to $\bar{6}$ under α. Since 6 can be written as

$$6 = 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0,$$

as a byte G can be represented as 00000110. Furthermore, since this byte can be represented as

$$0 \cdot x^7 + 0 \cdot x^6 + 0 \cdot x^5 + 0 \cdot x^4 + 0 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x^1 + 0 \cdot x^0 = x^2 + x,$$

as a polynomial G can be represented as $x^2 + x$.

Likewise all the elements in L can be represented as polynomial representations as shown below.

Letter	Polynomial representation	Letter	Polynomial representation
A	0	N	$x^3 + x^2 + 1$
B	1	O	$x^3 + x^2 + x$
C	x	P	$x^3 + x^2 + x + 1$
D	$x + 1$	Q	x^4
E	x^2	R	$x^4 + 1$
F	$x^2 + 1$	S	$x^4 + x$
G	$x^2 + x$	T	$x^4 + x + 1$
H	$x^2 + x + 1$	U	$x^4 + x^2$
I	x^3	V	$x^4 + x^2 + 1$
J	$x^3 + 1$	W	$x^4 + x^2 + x$
K	$x^3 + x$	X	$x^4 + x^2 + x + 1$
L	$x^3 + x + 1$	Y	$x^4 + x^3$
M	$x^3 + x^2$	Z	$x^4 + x^3 + 1$

The total number of bytes that are possible is $2^8 = 256$. Operations of AES algorithm are performed on bytes. Specific bytes are represented by elements in the finite field $F = \mathbb{Z}_2[x]/(p(x))$, of order 8 where $p(x)$ is an irreducible polynomial of degree 8 over $\mathbb{Z}_2$. In this study the choice of $p(x)$ taken is $p(x) = x^8 + x^4 + x^3 + x + 1$.

When encrypting a message using AES algorithm initially each character of the plaintext message has to be stored by using its polynomial representation in a matrix, which will be considered as the plaintext matrix of the message.

A table named **"S-box"** is used in the AES algorithm to transform a given element in $F = \mathbb{Z}_2[x]/(p(x))$, in to another unique element in *F*.

Table 1. AES S-box

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	99	124	119	123	242	107	111	197	48	1	103	43	254	215	171	118
1	202	130	201	125	250	89	71	240	173	212	162	175	156	164	114	192
2	183	253	147	38	54	63	247	204	52	165	229	241	113	216	49	21
3	4	199	35	195	24	150	5	154	7	18	128	226	235	39	178	117
4	9	131	44	26	27	110	90	160	82	59	214	179	41	227	47	132
5	83	209	0	237	32	252	177	91	106	203	190	57	74	76	88	207
6	208	239	170	251	67	77	51	133	69	249	2	127	80	60	159	168
7	81	163	64	143	146	157	56	245	188	182	218	33	16	255	243	210
8	205	12	19	236	95	151	68	23	196	167	126	61	100	93	25	115
9	96	129	79	220	34	42	144	136	70	238	184	20	222	94	11	219
10	224	50	58	10	73	6	36	92	194	211	172	98	145	149	228	121
11	231	200	55	109	141	213	78	169	108	86	244	234	101	122	174	8
12	186	120	37	46	28	166	180	198	232	221	116	31	75	189	139	138
13	112	62	181	102	72	3	246	14	97	53	87	185	134	193	29	158
14	225	248	152	17	105	217	142	148	155	30	135	233	206	85	40	223
15	140	161	137	13	191	230	66	104	65	153	45	15	176	84	187	22

The procedure of applying S-box transformation is expressed through the following example: Consider the polynomial $x^4 + x^2 + x$. It has the representation of the byte 00010110. The first and last four bits of this byte can be viewed as the binary expression of an integer between 0 and 15. In this procedure these integers indicate a specific row and a column of the Table 1. The first four bits, 0001 is the binary equivalent of the integer 1 and the last four bits, 0110 is the equivalent of 6. Therefore the value in 1st row 6th column of the Table 2.6.1 has to be selected, which is 71 and its binary equivalent is computed which is 01000111. The corresponding polynomial of this result is $x^6 + x^2 + x + 1$. Hence for the input $x^4 + x^2 + x$, the output of the S-box transformation is the polynomial $x^6 + x^2 + x + 1$.

The S-box transformation can be inverted by using another table similar to Table 2.6.1 or by reversing the steps of S-box transformation. AES encryption process include four layers namely *ByteSub (BS)*, *ShiftRow (SR)*, *MixColumn (MC)*, *AddRoundKey (ARK)* and the AES decryption process include the layers *InvByteSub (IBS)*, *InvShiftRow (ISR)*, *MixColumn (MC)*, *AddRoundKey (ARK)* which are the inverses of the AES encryption layers.

2.7. Keyed – Hash Message Authentication Code (HMAC)

A Keyed – Hash Message Authentication Code (HMAC) is a specific construction for computing a Message Authentication Code (MAC) which involves a cryptographic hash function in combination with a secret cryptographic key (Figure 4). As with any MAC it can be used to verify the *data integrity* and *authentication of a message*, simultaneously. Hash functions such as MD-5, SHA-1, SHA-256 etc. can be used to generate HMAC. In this study the secret cryptographic key was chosen to be the x-coordinate of the shared secret resulting from ECDH key exchange and hash function was chosen to be SHA-256. The result of HMAC can be identified as a signature of the sender of a message.

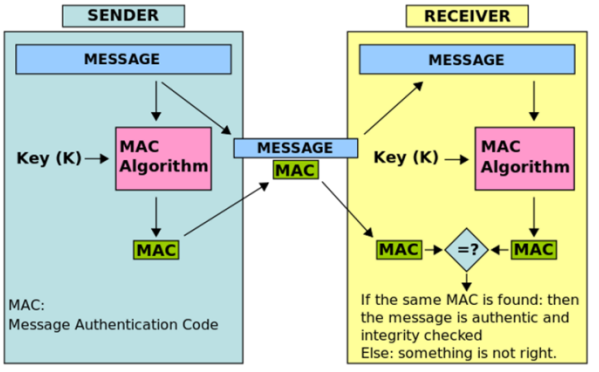

Figure 4. Process of a MAC

2.8. (m, 3m) - Repetition Code

Repetition code is one of the simplest error correcting codes found in Algebraic coding theory. The theoretical idea behind Repetition codes is to repeat the message several times when it is being transmitted over a noisy channel. For (*m, 3m*)-Repetition code, the encoding function can be defined as,

$$E : B^m \rightarrow B^{3m}$$

$$E(a_1, a_2, \dots, a_m) = (a_1, a_2, \dots, a_m, a_1, a_2, \dots, a_m, a_1, a_2, \dots, a_m)$$

Let $x, y, z \in B^m$. Then *xyz* denotes the word $w \in B^{3m}$ such that the first *m* letters of *w* are those of *x*, next *m* letters of *w* are those of *y* and the last *m* letters of *w* are those of *z*. Then the decoding function *D*, can be defined as follows:

$$D : B^{3m} \rightarrow B^m$$

The i^{th} digit of D(w), $w \in B^{3m}$, is the member that appears as the i^{th} digit in at least two of the words, x, y, z where $x, y, z \in B^m$ and w = xyz.
When constructing the codeword *w*, actually the encrypted word (say $x \in B^m$) is repeated thrice as *w = xxx* which can be denoted as *w = xyz* in general. The (*m, 3m*) - Repetition code principle can be extended to any *r* number of repetitions as (*m, rm*) - Repetition codes. In (*m, 3m*) - Repetition code, since the decoding is done by selecting the digit which occurs at least at two positions in each *x, y, z* blocks for each position, it is possible to detect two errors if present. But it can correct only single errors.

3. MATERIALS & METHODS

Since a cyclic subgroup of the abelian group of points on an elliptic curve is required to perform cryptography, initially a program together with a graphical user interface (GUI) was developed using MATLAB R2013a to illustrate the generation of cyclic subgroups, given any suitable elliptic curve defined over $\mathbb{Z}_p$. The GUI was designed in a manner allowing the user to input the parameters *a, b, p* of any elliptic curve $y^2 = x^3 + ax + b \ (mod p)$, defined over $\mathbb{Z}_p$, where *p* is a prime. For it to be an elliptic curve appropriate to be used for cryptographic purposes it should satisfy the conditions:

- The discriminent is non zero.
  $i.e. \quad \Delta = 4a^3 + 27b^2 \neq 0.$
- *p* is a prime.

Thus the program was developed to check the above two conditions initially within it and display an error message dialog box at a violation of any of the conditions. Thereafter in order to illustrate how security is achieved and implemented in ECC another program was written using MATLAB R2013a. There, to make the communication procedures clear for the scholars in the program Alice was selected as a sender of messages and Bob and Jane as receivers as chosen by Alice. After choosing an elliptic curve to perform the communication Alice can enter a message in the GUI designed, which will be subjected to an initial encryption known as the Advanced Encryption Standards (AES) and a secondary encryption based on the (m, 3m)-repetition code principles described in Algebraic coding theory in Abstract algebra. The secondary encryption is a novel concept introduced through this study as an improvement to the security offered by ECC.

This program included,

- Generation of a cyclic subgroup initially within it, given a suitable elliptic curve defined over $\mathbb{Z}_p$ chosen to perform a communication between two parties,
- User authentication between the two communicating parties due to a key exchange developed through the concept of Elliptic curve Diffie Hellman (ECDH) key agreement scheme ,
- Encryption and decryption of messages,
- Signature generation and verification by using keyed-Hash Message Authentication Code (HMAC).

A third GUI was designed to illustrate and implement the decryption of the received message by Bob or Jane, receival of Alice's HMAC signature and signature verification. At the receival of Alice's message in order to establish a successful user authentication the concept of ECDH was employed. A fourth GUI was developed to illustrate the additional security and advantage offered by the secondary encryption performed, if the message was intercepted and corrupted by a third party. Here it was assumed that there exists software technologies to identify the locations touched by a corruption source in the encrypted code and thereby the program was designed to identify the number of corruptions. It was also developed to rectify the corrupted message based on the locations of corruption and the number of corruptions present, utilizing the concepts in coding theory.

4. RESULTS & DISCUSSION

4.1. Results and Discussions related to Generation of Cyclic subgroups

Accomplishing the second objective of the study, a result of cyclic subgroups was obtained provided a suitable elliptic curve over $\mathbb{Z}_p$, where p is a prime. This result is intended to be highly advantageous for the analysis purposes of any scholar. It was observed that based on the generator of the cyclic subgroup chosen the cyclic subgroup generated varied.

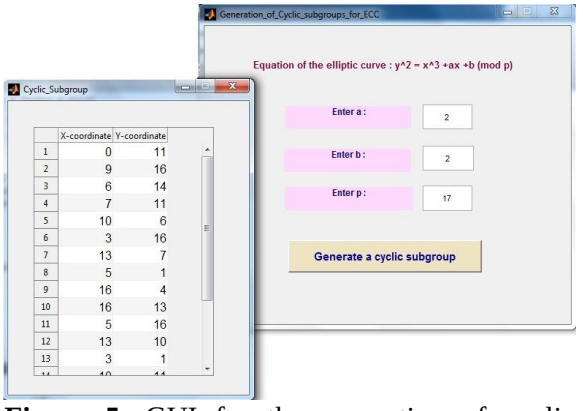

Figure 5. GUI for the generation of cyclic subgroups

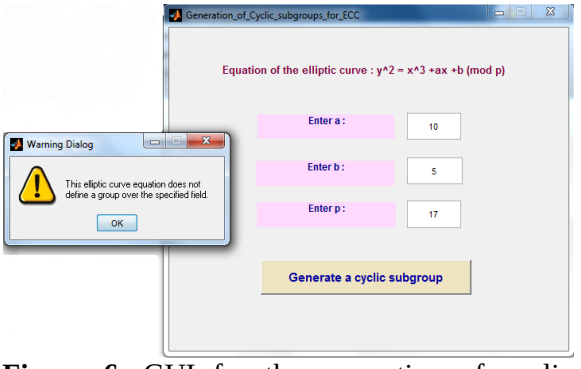

Figure 6. GUI for the generation of cyclic subgroups displaying error message for the violation of the necessary condition for the discriminant

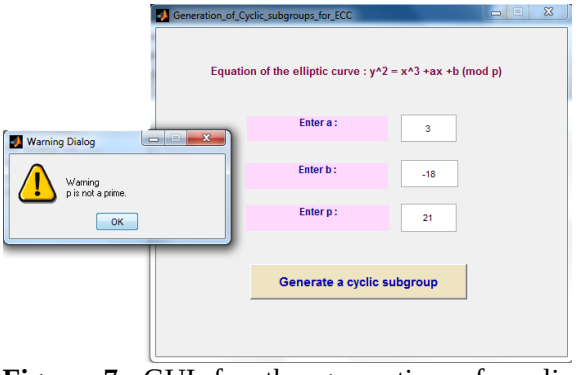

Figure 7. GUI for the generation of cyclic subgroups displaying error message for the violation of the requirement that p is a prime

4.2. Results and Discussion related to the Illustration of a Communication procedure under ECC

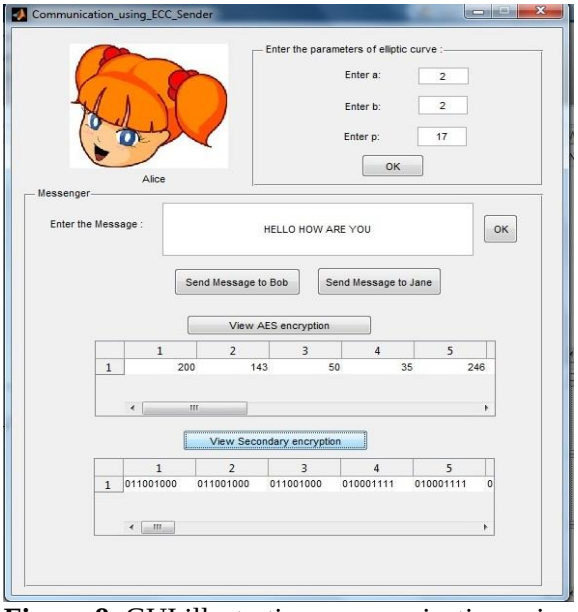

Figure 8. GUI illustrating communication using ECC-Sender

When illustrating the communication process using ECC, the program was designed to generate a cyclic subgroup based on the elliptic curve chosen to perform the communication. This also result in the two error messages created in the program for generation of cyclic subgroups, at the violation of any of the conditions necessary for a chosen elliptic curve to be a valid elliptic curve.

The interface allows the users to view the AES encryption and Secondary encryption of any given message as depicted in Figure 8. The program is designed to generate and store a HMAC signature for each message of the sender to be sent along with the message. To send the message the sender (Alice) can select among two receivers; Bob and Jane and at the selection made by Alice the user will be directed to a new GUI as shown in Figure 9.

The receiver chosen by Alice (i. e either Bob or Jane) is designed to receive a notification indicating the receival of the new message and also the state of user authentication; whether it is Successful or Unsuccessful. Moreover the ability to view Alice's HMAC signature and verification is also provided.

To illustrate the situation created if an interrupter (known as Eve in this program) interrupt and corrupt the message sent by Alice, two new separate GUI's were designed for Bob and Jane. At the click of the button "View Eve" in the respective receiver either GUI as shown in Figure 10 or Figure 11 will be opened.

The GUI's illustrating the communication interruption procedure clearly depicted the additional security and advantage offered by the secondary encryption performed. Due to the assumption that there exists software technologies that have the ability to detect locations touched by the corruption source it was possible to identify whether the message can be rectified or not through this program which is not possible according to the general theoretical concept of (*m, 3m*) – Repetition code.

According to the general definition for the (*m, 3m*) – Repetition code, if the same error occurs on two *x's* of a code word present as *w = xxx*, it will be decoded as a correct code since the same digits will appear in at least two locations of the repetitions which will result in an erroneous message. But according to the assumption made it is possible to identify that two *x's* are touched and corrupted and if two of the *x's* are corrupted a notification will be displayed to the user implying that it is *"not possible to rectify the errors"*.

On the other hand, for each section of the code word present in the form *w = xxx*, it was possible to correct any number of corruptions made to a single *x*. The number of corruptions present and the fact that it is possible to rectify the errors will be displayed to the user which is also offering additional convenience than the general (*m, 3m*) – Repetition code.

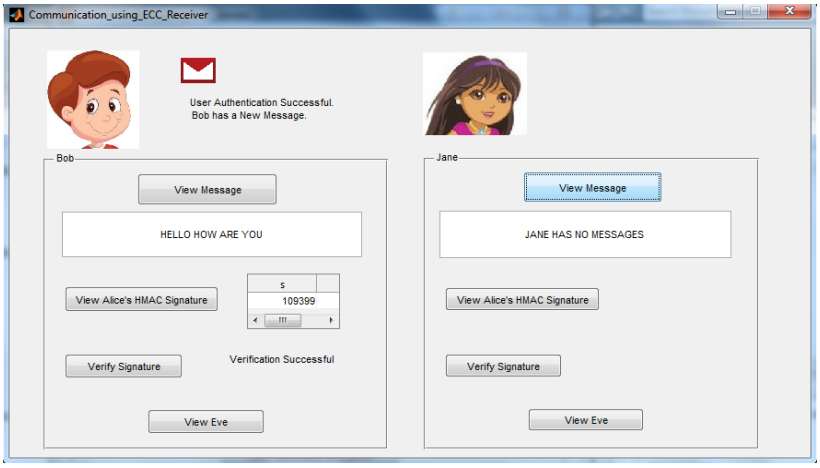

Figure 9. GUI illustrating communication using ECC-Receiver

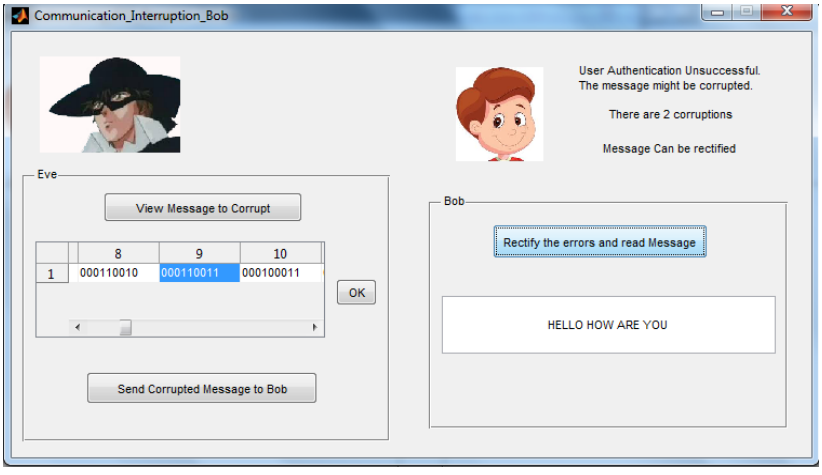

Figure 10. GUI illustrating communication interruption-Bob

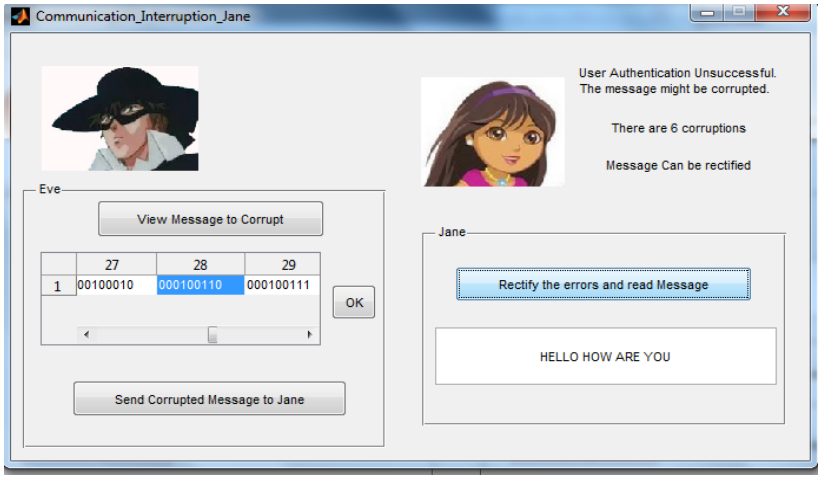

Figure 11. GUI illustrating communication interruption-Jane

5. CONCLUSIONS

Elliptic curve cryptography is the more advantageous cryptosystem at present with smaller key sizes and higher level of security.

It was evident that the basic requirements for a successful cryptosystem namely, confidentiality, data intergrity, user authentication and non-repudiation are actually met by ECC.

ECC is an aggregate of various disciplines such as algebra, number theory, computer science etc.

In order to improve the security offered by ECC it is possible to make use of mathematical concepts of algebra.

REFERENCES

AU S, TURNER CE & EVERSON J. The McEliece Cryptosystem. 2003.

KLIMA RE, SIGMON NP & STITZINGER EL. Applications of Abstract Algebra with Maple and MATLAB, Second edition, Chapman & Hall/CRC. 2007.

http://andrea.corbellini.name/2015/05/23/elliptic-curve-cryptography-finite-fields-and-discrete-logarithms/

https://en.wikipedia.org/wiki/Hash-based_message_authentication_code

KOPPENSTEINER C. Mathematical foundations of elliptic curve cryptography.

MALIK DS, MORDESON JN, SEN MK. Fundamentals of Abstract Algebra, McGraw-Hill Companies Inc. 1997.

MARTIN T. Elliptic curve cryptography Introduction. https://translate.google.com/translate?hl=en&sl=fr&u=https://www.smalsresearch.be/elliptic-curve-cryptography-tutoriel1/&prev=search

SILVERMAN JH. An Introduction to the Theory of Elliptic curves. 2006.

RIVEST RL, SHAMIR A & ADLEMAN L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. 1977.

SHUKLA AK & KAPOOR V. Data Encryption and Decryption using Modified RSA Cryptography Based on Multiple Public Keys and ‘n’ Prime Number, International Journal of Engineering Sciences & Research Technology. 2004.