Volume 66, Issue 4, 2022
Journal of Scientific Research
of
The Banaras Hindu University

Koblitz Curve- A Mapping Technique to Encipher/Decipher the Text message using Elliptic Curve Cryptography

Deepika Bhatia

Assistant Professor, Department of Information Technology,
Vivekananda Institute of Professional Studies, New Delhi, India, Email: deepika.bhatia@vips.edu

Abstract: **Elliptic Curve Cryptography (ECC) is studied to be very powerful and secured asymmetric technique which uses small key sizes and efficiently uses power and bandwidth. Wired and wireless sensor networks are prone to attacks like node capture, physical tampering, eavesdropping, denial of service, etc. ECC curves being resistant to these kinds of attacks, are used for secured data transmission. ECC uses Koblitz's curve mapping methodology for encoding and decoding of user's information over public environment and thus is efficient for secured computations. Koblitz curves are also used for secured exchange and ownership of cryptocurrency which is very popular nowadays. The research paper aims to discuss and implement Koblitz mapping technique on a given ECC curve and it is also proved to be secured to encode and decode the textual information during data transfer. The method, if used with parallel multipliers, does faster computations than other curves and saves time and space.**

Index Terms: **Encryption, Security, Data, Cryptography, Privacy, Encoding, Decoding etc.**

I. INTRODUCTION

ECC, elliptic curve is defined by the equation given below:

$$y^2 = x^3 + a*x + b$$

where, $4a^3 + 27b^2! = 0$.

Neil Koblitz and Victor Miller (1985), invented ECC curve. It is an asymmetric encryption technique which used public, private key pairs. This curve is based upon discrete logarithm problem that states that if we have two points P and Q on this curve such that Q=k*P, i.e Q is a product of scalar entity k and point P on this curve, then finding such value of k to satisfy this equation is very hard problem.

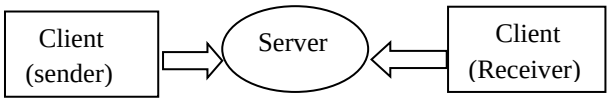

Fig.1: Client-Server communication

In case of RSA or DES algorithms, a client (for example browser) sends its public key to the server and requests for some data. The server encrypts the data with the help of client's public key and sends this encrypted data. And now the client receives this data and decrypts it. Fig. 1 above shows the scenario.

Privacy preserving schemes such as RSA, Paillier, DES and block cipher etc., face sever drawbacks as the cloud needs to decrypt this data before performing any computation over it and again this raises issue over the user's security. So, we need a scheme that allows us to do processing over this encrypted data without decrypting it over the service provider side. The algorithms which exhibit mixed homomorphic properties (they allow addition/ multiplication done any number of times) are best ones. ECC is one of such schemes which can be used for mixed homomorphic encryption property.

In case of ECC technique, when client requests for some data by sending his public key over the cloud, then sender will encrypt his data using his secret key (generated via ECC method) and receiver will decrypt it using his secret key (made from public key via this technique). So, security gets increased. ECC technique

DIO: 10.37398/JSR.2022.660417

shows same performance of work done by some of the researchers but uses shorter key size as compared to RSA or DES etc. This algorithm is good for devices having costlier hardware such as IOT devices. This method increases the numbers of browsers in same time intervals as compared to other security algorithms.

II. LITERATURE REVIEW

Rajeev Kumar et. al. (2018), proposed an improvement in Elliptic curve cryptography technique by overcoming the weaknesses in the scheme provide by Chaudhary et. al. (2015), which was vulnerable to an attack named impersonate. They tried to add authentication and security by encryption in the message. The author tried to overcome different types of attacks such as replay, man in the middle etc. Man in the middle attack failed using ECC by selecting random global parameters of the curve with which intruder tries to solve ECDLP problem. A secured e-payment scheme was proposed. An intruder, tries to do analysis of encrypted message by exploiting some features of ciphertext and tries to recover the key by applying various approaches to identify the algorithm working behind that. The author also studied brute force attack based upon finding the public key to obtain private key, but ECC proved to be more secured.

D. Sravana et. al. (2012), mentioned that ECC technique provides same level of security as RSA does and also with advantage of smaller key size. They tried to implement ECC technique over finite fields. ECC method provide a secret mechanism using Diffie Hellman key exchange method embedded in it for sharing the secret keys between two parties, and also authentic session key establishment protocols, Mohsen Machhout et.al., (2010).

M. Amara et. al. (2011), presented the basic mathematics behind ECC curve and also gave the justification that ECC has better advantages over RSA encryption algorithm over security aspects. Diffie Hellman elliptic curve algorithm is discussed by the author. Description of various modifications of ECC algorithms present in the market is given by the authors. Theory concept about security and performance of ECC is presented in this paper.

O. Reyad et. al. (2016), suggested about the scalar multiplication concept of ECC points. Addition and doubling operations of the curve generator point G is done with the help of scalar multiplication by multiplying G value with scalar factor k, a positive integer, given as: [k]*[G], where, G is defined by curve coordinates (x,y) and is known as generator point on the curve agreed upon by the users. This multiplication operation is defined as repetitive addition of G into itself. This value of k defines the strength of our ECC algorithms as it is one way operation and also finding the value of k*G is itself a tedious task. This reverse operation of finding the value of k, and then k*G is known as the Elliptic Curve Discrete Logarithm Problem (ECDLP) and is considered the core hardness of ECC. This value is very hard to find and gives the strength to this algorithm as this is a one-way function.

M. B. Yassein (2017), presented the comparative study of various symmetric and asymmetric data security algorithms. The researchers studied the difference between techniques such as AES, DES, 3DES, Blowfish, RSA, Diffie-Hellman and ECC. It has been stated that ECC technique is much efficient and more secured compared to other asymmetric methods. On the other hand, the symmetric methods prone to various attacks and limited scalable. The observations have been given in the following Table1.

Boni et. al. (2015), developed an improvement over Diffie-Hellman key exchange method and named it as multiplicative key exchange algorithm. It has been proved that this new method is faster than DHKE technique. It used simple and lesser mathematical computation behind key generation and also improved the speed of Diffie-Hellman key generation method.

Verma et. al. (2011), evaluated the performance of various encryption algorithms such as DES, 3DES, Blowfish and AES). They used different settings to evaluate encryption. The researcher implemented the code in C++ and .Net on a Pentium- 4, 2.1 GHz processor under Windows XP SP1. Processing time, key sizes, number of users i.e different load situations. It has been observed that Blowfish had stronger key size of 448 bits and has less processing time on various data sizes.

Kumar et al (2016), compared different encryption algorithms and observed that Blowfish gives the best results in various block cipher modes with minimum weak points. AES showed weaker performance compared with other algorithms. In general, symmetric encryption algorithms are faster than asymmetric encryption algorithm but it had only one weak point that it is shared its key with other parties involved in the process. Asymmetric encryption has a strength point that it is used two different keys but it's required more processing time than those in symmetric encryption.

III. PROPOSED METHODOLOGY

Elliptic curve cryptographic algorithm is based on the concept of elliptic curves. When user encrypts data with the help of ECC algorithm, the data is in secured form over the public cloud environment. Various users can share their encrypted data on the cloud, which can be later accessible in decrypted form to the other uses who request

Table1: Comparison of various Encryption Techniques

Encryption Method	**Factors/Parameters →** **Techniques**	**Key length**	**Speed (Encryption /Decryption)**	**Scalability and Attack Type**
*Symmetric	AES (By Joan and Vincent,1997)	128,192, 256	Faster	Not scalable and brute force attacked.
	DES (IBM,1975)	56 bits	Slow	Scalable and brute force attacked.
	3DES (IBM,1978)	168 bits	Very slow	Limited scalable and attack possible
	Blowfish (Bruce Schneier, 1993)	32- 448 Bits	Fast	Scalable and dictionary attack.
*Asymmetric	RSA (Rivest, Schamir and Andleman,1977) and based on integer factoring over prime numbers	1024 bits	CPU and memory intensive	Not scalable and brute force, plaintext and Oracle attack. Less Secured.
	Diffie-Hellman, (Wittfield Diffie, Martin Hellman, 1976) and based on discrete logarithm over finite fields	2013,224 Bits for q and 2048 bits for p	CPU intensive	Scalable and DOS attacked. Less secured.
	Elgamal is based on discrete logarithm over finite fields	Longer key size	CPU intensive	Secure and scalable.
	ECC (Given by Koblitz and Miller, 1985) based on discrete logarithm over finite fields	112 - 512 bits, shorter key size	Good for CPU and memory usage.	Scalable and timing or quantum computing attack. More secure

access. A public key is shared amongst two users to encrypt or decrypt the secured data sent over the public domain. The algorithm for exchanging the keys is named as Diffie Hellman key exchange algorithm using the concept of Elliptic curve points.

The Koblitz's mapping technique is explained in this paper and implemented for alphabets. Koblitz curve uses fast and complex computation with one way. Secp256k1 is proved to be secured in bitcoin currency transfer and unlikely to be attacked or hacked by intruders. Encoding and decoding method is illustrated with the help of an example. The ECC basic concept is given as follows:

A. ECC Security Technique:

1. Let two users Alice wants to share an encrypted message to the user Bob via a public media such as cloud.
2. For this process, both agree on some elliptic curve
3. $E_q(a,b)$, parameter choices. For the curve, domain parameters are:

{p, a, b, G, n, h}

p: it specifies a prime number which tell about the finite fields over the curve.

a and b: these are curve coefficients that helps to generate the curve.

G: is a generator point on the curve that generate a cyclic group reached by G. G is a global point on the elliptic curve whose order is large value n.

n: is order of G

h: is a cofactor and tell us about no of points on the curve.

4. Now users chose their own private keys and generate their public key pairs using some mathematical equations.
5. Both users now calculate their secret keys.
6. Now the sender can encode the plaintext message 'm' onto the given curve from an elliptic group. And send this cipher text onto the public media.
7. Receiver now decodes this point using his own calculated secret key using some mathematical computations given below and can decode the message 'm'.
8. By this method data security is ensured and security level is reached using lesser computation power and shorter key lengths.

B. Method to select Curve Parameters for Koblitz's Technique:

The order of the group in Koblitz curve has a large prime

factor. They are non-super-singular (so reduced attacks due to finite fields), and mathematical computation of curve points can be done using doubling method. Bitcoin also used Elliptic curve cryptosystem and has implemented many ECDSA algorithm for secured transaction on the communication network.

In Koblitz's algorithm, the maximum possible value for 'm' is 128, if an 8-bit number is encrypted. For the value of k=20, the minimum value of x is m*k+1 ie 128*20+1=2560+1 to represent a character. To get a point on the curve whose x-coordinate is greater than 2560, we need to select an elliptic curve with p value not less than 2560. So, depending on the value of k (>=20) we need to select the curve parameters.

C. *Koblitz's Method to find the message point onto the given Curve:*

1. *Encoding Plaintext message 'm':*

1. Consider an elliptic curve Ep(a,b) for some values of p, a and b parameters.
2. Now generate number of points on this curve E. let n points are generated.
3. For user's data, we can consider alphabets-Capital and small both and also digits from 0 to 9 to encode on this curve Ep(a,b).
4. As per the algorithm of Koblitz, value of an auxillary base parameter 'k' is to be chosen such that it should satisfy the following constraint:

$$[(m+1)*k]<p$$

Both parties should agree on this value of k.

5. Message 'm' is represented as

$$x=mk+j \quad , \text{where } j=1 \text{ to } k-1$$

6. For j=0,1,---,k-1, find the value of equation $x^3 + ax + b$ and try to find the solution for 'y'.
7. If we find the value of y for this solution, then it is ok else try for another value of j and find y.In practice, you will find such a *y* before you reach upto the value $x = mk + k - 1$.
8. Now consider the message as this paired value of (x,y) and consider it as a point on the given curve.

2. *Decoding the plaintext message 'm':*

1. Now find the value of m1=(x-1)/k and take the value of m to be the greatest integer such that it is <= m1. So the point (x,y) is decoded as the message m.

Example: Say the parameters of curve are:

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

In this case, prime number is 1213 and also 'n' is number of points generated on this curve through the given generator point. The word to be encoded is given as GOD. The following procedure shows us how to calculate the encoded value of G, O and D characters step by step using the above curve base parameters.

1. Let us take character 'G' to be sent to other party.
2. 'G' is first encoded as number 7.
3. Let us take the value of auxillary parameter k=20 for this curve.
4. Now computer the value of x=mk+1 i.e 7*20+1=141 and try to solve it for a y such that y2= x3 + a*x+ b mod p .
5. So now find next value as we didn't find any value of y in previous step. Thus x=7*20+2 i.e x=142, and solve it for y if exists. We found the y value for this x as y=311.
6. Now the point (142,311) is point is encrypted and decrypted as a message.
7. To decode just compute (x-1)/k ie (142-1)/20=141/20 i.e 7.05.
8. Return number '7' as original plaintext as it is the greatest integer less than (x-1)/k, that is 7.
9. The computed digit 7 is now decoded to character 'G'.
10. Similarly, we can encode and decode the remaining character 'O' as (301,301) and 'D' as (81,465).
11. The probability that we fail to find a square (and thus fail to associate m to a point [14]) is about 1/2k [10].

IV. IMPLEMENTATION RESULTS

1) *Output 1: Number of points and iterative value generations on the curve:*

Following figure 2 shows the output showing total numbers of points that can be generated on the given elliptic curve having curve like:

$y^2 = x^3 + 2x + 2$ (modulo 17)

```
Enter the values a and b=2
2
Enter the value of p where p should be a prime number:=17

The points generated for the curve are (x,y):= 0,6    0,11   3,1    3,16
5,1    5,16   6,3    6,14   7,6    7,11   9,1    9,16   10,6   10,11
13,7   13,10  16,4   16,13
 Total number of generated points (+1 including infinity) are:=19
```

Fig.2: Number of Generated Points

For the above curve itself, the following output shows that if we chose the Generator parameter G, as = (5,1), then multiplication of 17*G can be carried using repeated additive property of Elliptic curve, after some complex modulo mathematical computations. The inverse of the given generator point is calculated as (6,14) and is shown in figure 3 below.

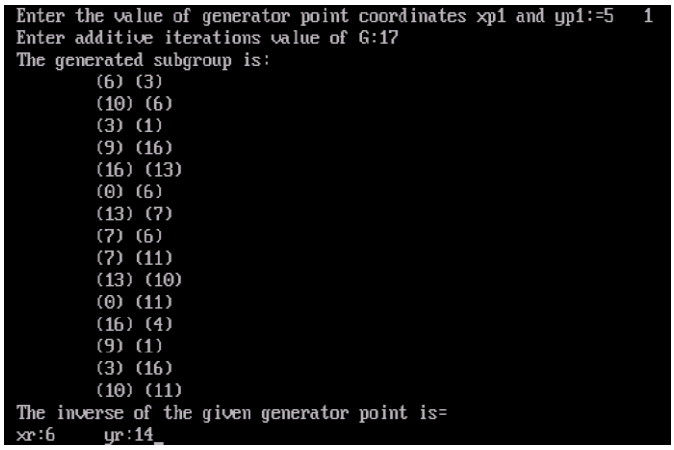

Fig.3: Inverse of Generator point G

If for the same curve additive iterations for G is 18, then the inverse of given generator point (5,1) is given by (5,16). The figure 4 below shows various coordinate pairs as the finite fields that can be generated by using the input parameter and also shows the inverse using given generator point for the given curve modulo 17.

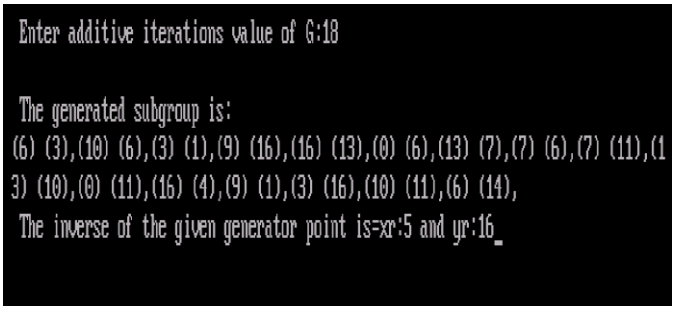

Fig.4: Finite group and inverse of Generator point G

2) *Output 2: Koblitz's Method to find the message point onto the given Curve and for encoding the Plaintext message 'm':*

Mapping for Alphabets is given as shown in the table 2 below. The points on the specified Elliptic curve are shown below as in the mapping table. The following data shows the ASCII codes of capital alphabets. So, with the help of encoding done by above Koblitz method the ASCII code is distributed uniformly on the given curve. So, this shows a good uniform property distribution of points on the given curve E.

3) *Output 3: Encoding of alphabets on the curve Ep(a,b) is given below:-*

- *Encoding of the word GOD:*

In the following figure, the word GOD is encoded. The alphabets are encoded as (x, y) coordinates pair on the given curve. For example:

'G' alphabet is encoded as (142,311)
'O' alphabet is encoded as (301,301)
'D' alphabet is encoded as (81,465)

The parameters chosen for this curve are a=b=2, p=1213 and n=1219, G= (37, 47). The figure 5 shows the encoding of given message using the curve parameters. Similarly, we can encode other words also using the ECC character mapping methodology.

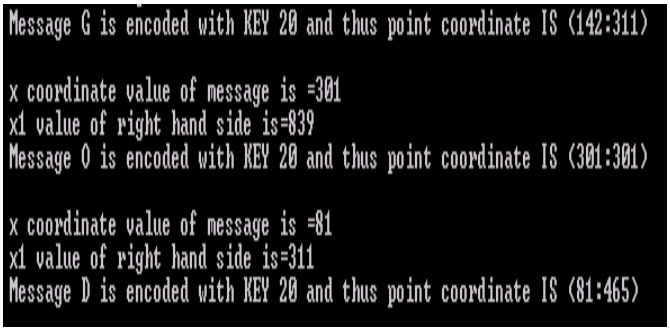

Fig.5: Encoding of Message

4) *Output 4: Decoding of alphabets on the curve Ep(a,b) is given below:*

- *Decoding of the word GOD shown below:*

In the following figure, the word GOD is decoded. The alphabets are decoded as (x, y) coordinates pair on the given curve. For example:

'G' alphabet is encoded as (142,311) is decoded now.
'O' alphabet is encoded as (301,301) is decoded now.
'D' alphabet is encoded as (81,465) is decoded now.

Table2: Mapping for Alphabets

Mapped Coordinates (x,y) for Alphabets						
S	**A**	**B**	**Symbols with ASCII code and Encoded (x, y) coordinates value and curve parameters given as $y^2 = x^3 + 2x + 2$ (modulo 1213) G=(37,47), Private Key=3**			
E	(21,381)	(41,77)				
AC	65	66				
S	**C**	**D**	**E**	**F**	**G**	**H**
E	(61,525)	(81,465)	(101,376)	(121,243)	(142,311)	(161,260)
AC	67	68	69	70	71	72
S	**I**	**J**	**K**	**L**	**M**	**N**
E	(181,31)	(201,110)	(225,528)	(242,229)	(261,174)	(282,316)
AC	73	74	75	76	77	78
S	**O**	**P**	**Q**	**R**	**S**	**T**
E	(301,301)	(322,229)	(347,303)	(361,20)	(382,106)	(401,249)
AC	79	80	81	82	83	84
S	**U**	**V**	**W**	**X**	**Y**	**Z**
E	(425,448)	(441,576)	(461,39)	(481,553)	(502,119)	(522,534)
AC	85	86	87	88	89	90
S*- Symbol, E*- Encoded coordinates x and y, AC*- ASCII code value						

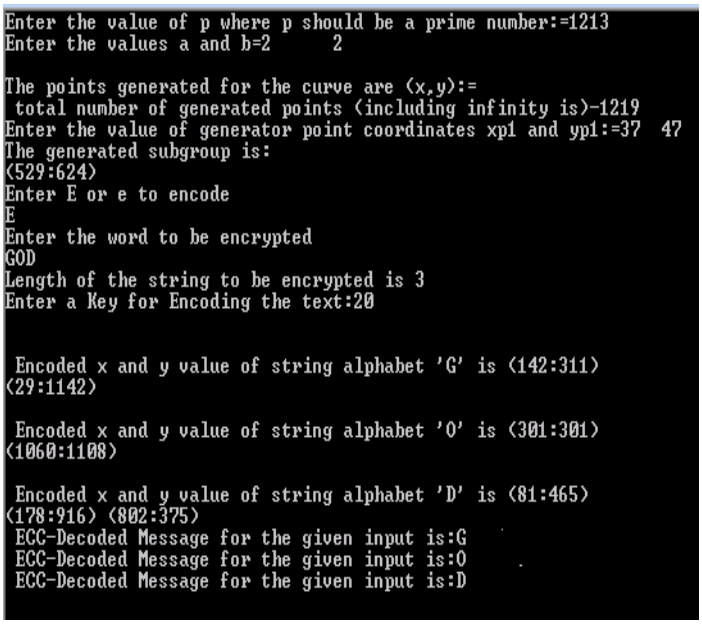

Fig.6: Decoding of Message

Figure 6 above shows the decoding of user's message on the given elliptic curve. Similarly, we can decode other words also using the ECC character mapping methodology. Here the encoding key used to encode the message 'GOD' is chosen and agreed upon as k=20. So, message has been encoded and decoded successfully on the given elliptic curve with finite field 1219.

V. Conclusion and Future Work

In this paper a mapping methodology is implemented and results are shown. This technique works on ASCII code of user's data. The data sent by the encoder is encoded as a pair of (x,y) coordinate values and later decoded on the receiver side. This method basically shows the uniform distribution of data on the specific elliptic curve. It has been studied that the users achieve better security using ECC mapping technique called Koblitz method. To conclude that, the technique has been implemented successfully and also can work on differently sized curves. Data security is attained as user's data is secured as encoded pair on given curve E. The paper also analyzed the work proposed by various researchers on concerns like security, performance, cost and various parameters.

In future work, we would implement the encryption and decryption of these encoded and decoded points using Elliptic curve encryption-decryption algorithm. This paper took user's alphabetical data into consideration and in future, work will be done on numeric data as well.

ACKNOWLEDGEMENT

I would like to thank to the anonymous reviewers for their

valuable and helpful comments that are greatly helpful to improve the methodology and thus quality of this work.

REFERENCES

Rajeev Kumar, S.K.Pal and Arvind Yadav (2018), "Elliptic curve based authenticated encryption scheme and its application for electronic payment system", Int. J. Computing Science and Mathematics, Vol. 9, No. 1, pp. 90-101.

Chaudhry, S.A., Farash, M.S., Naqvi, H. and Sher, M., (2015) "A Secure and Efficient Authenticated Encryption for electronic Payment Systems using Elliptic Curve Cryptography", Springer Science and Business Media, New York, pp.113–139.

D. Sravana Kumar, Ch. Suneetha and A. Chandrasekhar, (2012) "Encryption of Data Using Elliptic Curve Over Finite Fields", International Journal of Distributed and Parallel Systems (IJDPS) January, Vol.3, No.1, pp. 301-308.

K R Chandrasekhara Pillai1 and M P Sebastian,"Elliptic Curve based authenticated session Key establishment protocol for High Security Applications in Constrained Network environment", International Journal of Network Security & its Applications (IJNSA), Vol.2, No.3, July.

Kin Choong Yow and Amol Dabholkar, (1985) "A Light-Weight mutual authentication and key-exchange protocol based of elliptical curve cryptography for energy-constrained devices, International Journal of Network Security & its Applications Vol. 2 No. (CRYPTO 1985), Springer LNCS 218, pp. 417-4 26.

Mohsen Machhout et. al., (2010) "Coupled FPGA/ASIC implementation of elliptic curve crypto-processor", International Journal of Network Security & its Applications Vol. 2 No. 2 April.

Ch. Suneetha, D. Sravana Kumar and A. Chandrasekhar, (2011) "Secure key transport in symmetric cryptographic protocols using elliptic curves over finite fields", International Journal of Computer Applications, , Vol. 36, No. 1 November.

Darrel Hankerson, Alfered Menezes, Scott Vanstone, (2004) "A Guide to elliptic curve Cryptography", Springer.

V.Miller, (1985) "Uses of Elliptic curves in Cryptography".In advances in Cryptography (CRYPTO 1985), Springer LNCS 218, pp. 417-426.

Omar Reyad, (2018) "Text Message Encoding Based on Elliptic Curve Cryptography and a Mapping Methodology", Inf. Sci. Lett. **7**, No. 1, pp. 7-11.

T.N. Shankar and G. Sahoo, (2009)Cryptography with Elliptic Curves, Int J of Computer Science and Applications 2, pp. 38–42.

K. Agrawal and A. Gera, (2014) Elliptic Curve Cryptography with Hill Cipher Generation for secure Text Cryptosystem. Int J of Computer Applications 106, pp. 18–24.

M.C. Vigila and K. Muneeswaran, (2009) "Implementation of text based cryptosystem using elliptic curve cryptography", ICAC IEEE 10, pp. 82–85.

Padma Bh et. al., (2010) "Encoding And Decoding of a Message in the Implementation of Elliptic Curve Cryptography using Koblitz's Method", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 05, pp. 1904-1907.

Moncef Amara and Amar Siad, (2011) "Elliptic Curve Cryptography And its Applications", 7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA), pp. 247-250.

N. Koblitz, (1987) "Elliptic Curve Crytosystems," Mathematics of Computation, Vol.48, pages 203-209.

V. S. Miller, (1985) "Use of Elliptic Curves in Cryptography," Advances in Cryptology - CRYTO '85, Lecture Notes in Computer Science, vol. 128, Springer- Verlag, Hugh C. Williams (Ed.), pp. 417-426.

O. Reyad and Z. Kotulski, (2016) "Pseudo-Random Sequence Generation from Elliptic Curves over a Finite Field of Characteristic", In: Federated Conference on Computer Science and Inf. Sys., FedCSIS, ACSIS 8, IEEE, pp. 991–998.

M. B. Yassein et.al., (2017) "Comprehensive Study of Symmetric Key and Asymmetric Key Encryption Algorithms" , ICET2017, Antalya, Turkey, 978-1-5386-1949-0/17, IEEE, pp. 1-8.

Boni, Sharad, Jaimik Bhatt, and Santosh Bhat, (2015) "Improving the Diffie-Hellman Key Exchange Algorithm by Proposing the Multiplicative Key Exchange Algorithm." International Journal of Computer Applications 130.15.

Verma, Om Prakash, et al. (2011) "Peformance analysis of data encryption algorithms." Electronics Computer Technology (ICECT), 3rd International Conference on. Vol. 5. IEEE.

Kumar, Praveen, et al. (2016) "A performance based comparison of various symmetric cryptographic algorithms in run-time scenario." System Modeling & Advancement in Research Trends (SMART), Internationa Conference. IEEE.

Simrandeep Singh Thapar et. al., (2018) "A Study of Data Threats and the Role of Cryptography Algorithms", 978-1-5386-7266-2/18, pp. 819-824, IEEE

Jerome A. (March 2020) "Solinas Efficient Arithmetic on Koblitz Curves", National Security Agency, Ft. Meade.

The Bitcoin Wiki, Secp256k1. (2015) [https:// wiki.bitcoin.com /w/ Secp256k1] October 31.

SafeCurves, Index. [http://safecurves.cr.yp.to/index.html].

Bitcoin Talk, forum. Dan Brown (2013) e-mail reply. [https:// bitcointalk.org/index.php?topic=289795.msg3183975msg3183975] September 18.

PCWorld. (2014) Overreliance on the NSA led to weak crypto standard, NIST advisers find. [http://www.pcworld.com/ article/ 2454380/ overreliance-on the-nsa-led-to-weak-crypto-standard-nist-advisers-find.html] July 15.

Daniel J. Bernstein, Tanja Lange. Security dangers of the NIST curves.[http://www.hyperelliptic.org/tanja/vortraege/ 20130531.pdf

***