Dr. Nagaratna P. Hegde, et. al. International Journal of Engineering Research and Applications
www.ijera.com
ISSN: 2248-9622, Vol. 12, Issue 4, (Series-IV) April 2022, pp. 35-40

RESEARCH ARTICLE OPEN ACCESS

Elliptic Curve Cryptography using Authenticated Encryption

Dr. Nagaratna P. Hegde [1], P. Deepthi [2]
[1] *Professor, Computer Science and Engineering, Vasavi College of Engineering, Hyderabad, Telangana, India*
[2] *Assistant Professor, Computer Science and Engineering, Bhoj Reddy Engineering College for Women, Hyderabad, Telangana, India*

ABSTRACT
Asymmetric encryption is used by many applications to provide secure communication between two parties. Asymmetric encryption uses more memory and require more computation. Elliptic Curve Cryptography (ECC) is an asymmetric cryptographic technique that is widely in use on small computational devices because it has the effect of using a strong cryptographic mechanism to generate small keys. ECC is used in a variety of devices, like sensors, Internet of Things (IoT), etc., [3], to reduce power consumption and improve device performance. ECC is strong to implement for the secure communication, if the information is encoded on an Elliptic curve. Equally important is ensuring that ECC maps the message on to the elliptic curve which can be used for encryption. The goal of this work is to provide authenticated encryption for encoding message and map the message on to the curve.

Keywords: Elliptic Curve, Elliptic Curve Cryptography, Authentication

---

Date of Submission: 12-04-2022 Date of Acceptance: 29-04-2022

I. INTRODUCTION

Asymmetric encryption is used by many applications to provide secure communication between two parties. Asymmetric encryption uses more memory and require more computation. Elliptic Curve Cryptography (ECC) [2] is an asymmetric cryptographic that is widely in use on small computational devices because it has the effect of using a strong cryptographic mechanism to generate small key size. ECC is used in a variety of devices, from sensors to Internet of Things (IoT) devices, to reduce power consumption and improve device performance. ECC requires a strong implementation to ensure secure communication, especially if the message is to be encoded on an elliptic curve [4]. Equally important is ensuring that ECC maps the message to the curve used for encryption. The goal of this work is to propose a reliable scheme that provides authenticated encryption for both message encoding and map it on to the curve.

Another name for public key cryptography is asymmetric cryptography, which uses two keys public and private. Public key is used by sender for encryption of the data. In other words, this key should be available to all parties. Private key is used to decrypt the encrypted data and only the recipient needs to know about this. This protocol solves the need for a secure key exchange between two parties [3]. However, there are drawbacks to using asymmetric encryption: like the size of the key and the required number of calculations.

Elliptic Curve Cryptography (ECC) is often used to solve key size and computational problems when low-end devices need to maintain performance. The amount of security provided by ECC is the same as RSA compared to the key size. Cryptography is at risk to many known attacks that jeopardize the cryptographic process.

II. MOTIVATION

The motivation behind this task is a security vulnerability that can make use of many attacks presented in the previous section. Many articles do not explain how to perform the encoding of the message on to a curve. Many security issues need to be understood for providing authenticated cryptographic encryption.

The core of this work is to provide a authenticated scheme using ECC as follows:

- Describes a vulnerability in the ECC to encode and map.
- Protect the message by applying a block cipher mode to encode that is resistant to some attacks.

- Provides study that have a significant impact on schema performance during the coding stage for padding steps.

III. LITERATURE SURVEY

ECC is used to exchange the data and protect the communication which is between the two parties. There are some techniques that have been proposed to reduce the computational effort required for finding the keys.

One way to encode a message is by using ASCII code, then convert it to bits and then to decimal, or to use a mapping table. They are not protected from many attacks, like plaintext attacks. Many techniques were invented to overcome these kinds of attacks, one of the remarkable methods is XOR dependent and which is mapped using a hidden matrix. They are not protected to ciphertext attacks or collision attacks. Many techniques do not provide authenticated encryption.

These days proposed techniques makes use of ECC to lessen cryptographic computations and conquer the restrictions of low-stop devices. Almost all those techniques do now no longer offer information on the way to convert a message to a range of and map it to an elliptic curve. These techniques offer many upgrades in particular areas further, to enhancing the safety of ECC itself. The relaxation of this literature overview makes a speciality of particularly how messages are transformed to numbers and mapped to the elliptic curve.

IV. BACKGROUND

ECC believes that its effectiveness is the future for protecting communications on small-computing devices such as IoT and sensors. Several techniques have been proposed here to use ECC for protecting nodes with poorly performing resources. However, these techniques have weaknesses in the implementation of ECC that suffer from serious security issues.

These restrictions are like using a different elliptic curve, which rely on the message encoding that is weak, and allowing the weak messages to map on the curves. In addition, most of the systems fail because they provide only integrity of encrypted messages. That is, it does not provide effective authentication. These considerations need to be considered when developing a technique that gives confidentiality and integrity to communicate from one another over the internet.

In addition, this technique needs to be validated for ensuring a robust level of security to the multiple cryptographic attacks, which include plaintext, ciphertext attacks. This technique works well for devices with less computing power and limited resources, and for minimizing processing time and memory capacity.

4.1 Elliptic Curve Cryptography (ECC)

It is used in less computing devices such as wireless sensor networks, Internet of Things (IoT). Because it offers almost equal level of strength as many other public key algorithms, has the advantage of using small key size and less computations. Example, RSA algorithm uses 1024-bit key for encryption will be equivalent to ECC encryption with a 160-bit key. As there is big differentiation in the size of the key it follows that less computing devices work better. Importantly, ECC depends on the discrete logarithmic for a given elliptic curve on a finite field. It uses to provide communication securely between the two parties for exchanging keys, signing messages for integrity, and providing the ability to prevent forgery. Many techniques use ECC for communicating securely, but these techniques differ in some areas of encryption.

As mentioned earlier, certain techniques are only used to facilitate key generation, others are used for both message encryption and signing. On the elliptic curve (F_q),

where $q > 3$ the set of pairs $(x, y) \in F_q$

The equation 1 gives the elliptic curve equation

$$y^2 = x^3 + a\,x + b \bmod q \quad (1)$$

where $a, b \in F_q$ and $4a^3 + 27b^2 \neq 0 \bmod q$. The elliptic curve must be non-singular. That is the graph has no self-intersections or vertices. Figure 1 shows an example of the elliptic curve $y^2 = x^3 - 4x + 8$

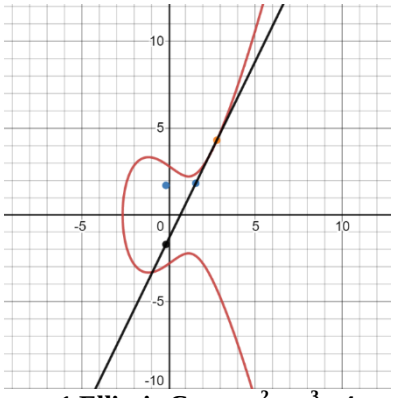

Figure 1 Elliptic Curve $y^2 = x^3 - 4x + 8$

Addition '+' is a group operation which can be represented on an elliptic curve.

Let $P = (x_1, y_1)$

$Q = (x_2, y_2)$.

Then the $P + Q = (x_1, y_1) + (x_2, y_2) = (x_3, y_3)$.

Figure 2 shows an example of point addition for elliptic curve.

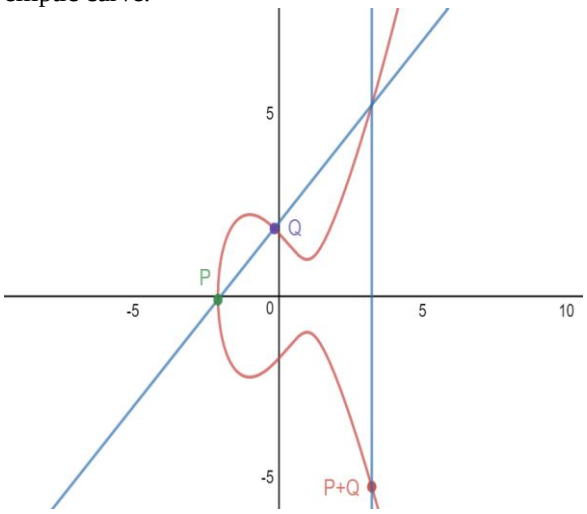

Figure 2 Point addition

If P = Q, then P + P = $(x_1, y_1) + (x_1, y_1) = 2P$.
2P is called point doubling.
The Figure 3 shows the example of point doubling.

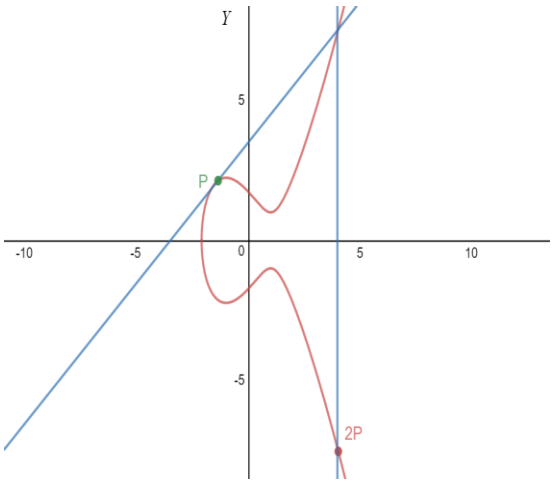

Figure 3 Point Doubling

The main operation of ECC is group multiplication. This is defined by counting the doublings of the group points. It consists of an integer called the secret key, x, and one base point of the elliptic curve, F = (M_i, N_i).

Therefore, the operation xF is called x-doubling time of F, which leads to another point (M_j, N_j) called the public key. ECC security is based on the hardness of the math problem, showing that if one knows the base point and the public key, one cannot find x in polynomial time. This is called the Discrete Logarithmic.

There are several stages in ECC for secure communicating of keys. Elliptic curve definitions, parameters such as private and public key calculations, generate message numeric coding (for encryption signature), and assignment of encrypted messages to elliptic curves.

The main calculation of the first ECC stage is calculating public key. ECC's public key is calculated as product of X (ie secret key) and F (ie, the base point). The secured public key is calculated using scalar multiplication of elliptic curves.

In the second stage the techniques are used for converting the characters in the message to numbers because the encryption of ECC is based on numbers. Each character in the message must be encoded for preventing from different kinds of attacks like the plaintext and ciphertext attacks(Figure 6). Similarly, the third stage verifies that the sender sent the message and signs the message to protect it from external changes.

In the final stage the message is mapped on to the elliptic curve. The mapping of message is important to prevent against various attacks. The encrypted message must be signed to protect authentication.

V. The Message Must Be Converted To Numbers

Various techniques offer approaches that allow messages to be numerically encoded (like ASCII) and elliptic curves mapped.

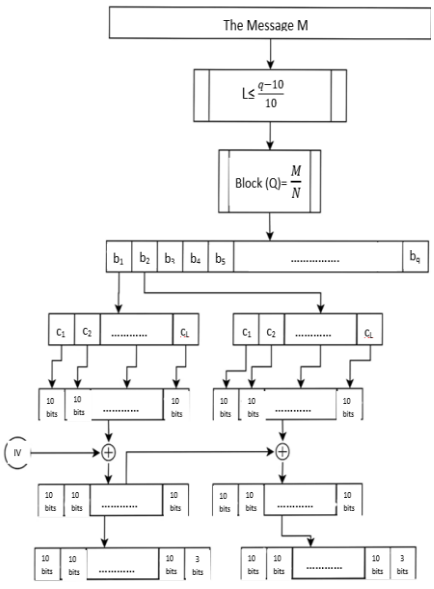

Figure 6 Shows Encoding of Message

ASCII table is known for everyone which is the main drawback of this method, to have plain text attack.

One need to ensure the integrity of encrypted messages. One also must ensure that the ciphertext is encrypted for authentication.

VI. MAP MESSAGE ON AN ELLIPTIC CURVE

The message must be mapped after encoding on to the curve to form $(M_i, N_i) \in Fq$ (a, b).

If a proper N_i is not found, M_i must be incremented with 1 till another N_i is found. Therefore, many techniques try to find the corresponding N_i and add a specific bit to the message to avoid M_i changes. Finding the right N_i is sometimes difficult. As a result, some techniques simply don't care of N_i, as there is no role for N_i in decrypting the message. Improper mapping can lead to security breaches if the technique provides encoding effectively. So, effective techniques provide practical and complementary encoding and mapping mechanisms. ECC encryption is grouped into encoding a message and mapping it on to the curve.

VII. SIGN THE ENCRYPTED MESSAGE FOR AUTHENTICATION

Confidentiality, integrity, and availability (CIA) comes under information security. Encryption guarantees the first part, but it cannot guarantee integrity by itself. One assigns a value to maintain integrity for message or ciphertext.

One way hash function is used for obtaining the value for calculating the message and this is sent to the receiver for verification. This process allows one to add the private key of the sender to the one-way hash function for the maintenance of the integrity of the message and for ensuring non-repudiation.

The receiver can verify the message sent by using senders public key not by using the private key. Signature only will not provide confidentiality, only it can prevent incoming messages from being tampered with by unauthorized users. ECDSA, a well-known method of signing messages on low-complexity devices, is effective to provide small keys (such as 160-bit keys) for the same level of signature(and security).

VIII. THE PROPOSED SCHEME

Authenticated techniques include converting messages to numbers, mapping encrypted messages to elliptic curves, encrypting messages with public keys, signing encrypted messages, and validating and decrypting messages with public keys. It consists of eight stages of decoding the message to numbers. This paper helps provide an ECC-based authentication scheme with secure message encoding, mapping, and encryption.

Along with these stages, the first stage is important given that many of the techniques consider the importance of sharing a common key between the two to encrypt the message. This stage is included here as it can reasonably be considered as an important stage of the system to facilitate authentication. Figure 4 shows the eight stages of the proposed system.

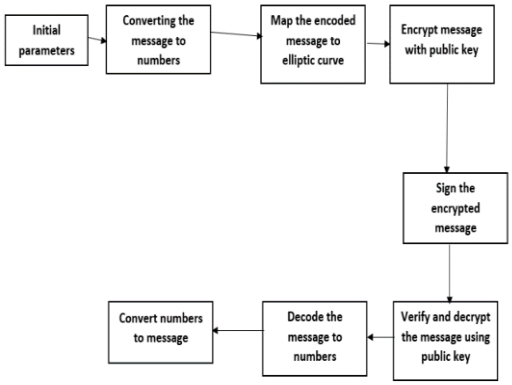

Figure 4 Overview of proposed scheme

8.1 Initial Parameters

The important advantage of this stage is that a shared private key is generated between the two parties. Elliptic curves use the key for encoding of points for mapping.

A shared session key is created by the sender for encryption of the points on the curve. Using the private key t_s and the recipient's public key $public_r$, the sender generates the session key_{sh}. The generation of the key is shown in Figure 5. Both the sender and the receiver in ECDLP allows to agree a common key.

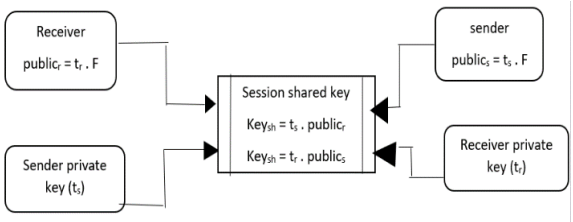

Figure 5: For Creating a Shared key

8.2 Encode the message to a number

This work extends the steps of the encoding and mapping mechanism to overcome the vulnerabilities in the present techniques. The message provided must be divided into separate blocks of B, where each block is having N characters each.

Calculate N using the following formula.

$$L \leq \frac{q-10}{10}$$

Therefore, in the authentication method, the value of L is 10. Where q = 112. The required Q (number of blocks) can be calculated as follows:
The following formula is used to achieve this

$$Q = \frac{M}{N}$$

The reason for splitting the message to this length is that each mapped point must be encoding on the curve, which should have the same size as q. The two leading characters are removed from each block for filling each of the block with the 3-bit of zeros which are required for the mapping stage. For each message, when a block of message M is received, the characters in the block are converted to their binary equivalent.

The first set of block binary values are XORed with the initial vector and followed by the subsequent blocks are XORed with the previous XORed blocks. Then the 3-bit XOR block is sent to the next stage.

8.3 Map the Message on to an Elliptic Curve

If a point (M_i, N_i) satisfies the elliptic curve given by Equation 1then map the message on to an elliptic curve. So, for every M_i value the corresponding N_i for each point must be calculated. At each block of encoded message, the block the binary value is converted to the decimal value. Using this the corresponding elliptic curve is plotted. Figure 7 shows the process of mapping an encoded message on to an elliptic curve.

8.4 Encryption of the Points for mapping

Many techniques assume that the mapping stage is sufficient to protect the message they don't look at the encryption stage. But, if the point is being mapped on to an elliptic curve, this assumption is incorrect because the secret key can be used to plot the point to obtain the ECDLP hardness. These points are protected in several ways. In the technique shown the points are encrypted by adding them to the session key key_{sh}. As a result, it is very difficult to get the points that are mapped on the curve without a knowing shared key. Figure 8 shows the steps involved in encryption of the points for mapping.

8.5 Signature

Authentication encryption scheme ensures the confidentiality and integrity of messages that is exchanged between two parties. This scheme maintains confidentiality during the stages. The sender signs the encrypted points to maintain integrity. All the points are mapped on the curve.

8.7 Verify Message Signed

The receiver decrypts use senders public key to validate the message.

8.8 Decrypt Message

Decryption is opposite of encryption. To decrypt the points that are encrypted, the receiver must use session key_{sh}.

8.9 Decode the Message that is Decrypted

In this stage the message is converted to its binary form.

8.10 Convert the Decoded Message back to Plaintext

In this stage the message is decoded back to the plain text.

IX. CONCLUSION AND FUTURE WORK

In this paper, an authentication scheme is introduced, an ECC-based authentication scheme encodes the message effectively and then map it to the elliptic curve. The proposed technique has the advantage of addressing the coding stage and is resistant to multiple cryptographic attacks. In addition, this work will perform evidence-based security analysis to shed light on the level of security of the proposed scheme. Describes the padding reduction properties of the scheme, along with some important output sets based on the well-known and proven elliptic curves. This task uses a variety of metrics to show that authentication techniques are superior to other techniques in terms of attack resilience, padding size, number of encoding operations, and number of decoding operations.

Future research can explore the implications of processing cryptographic properties at the mapping stage rather than the encoding stage.

REFERENCES

[1]. J. Guziur, M. Pawlak, and A. Poniszewska-Maranda, and B. Wieczorek, ''Light blockchain communication protocol for secure data transfer integrity,'' in Proc. Int. Symp. Cyberspace Saf. Secur. Cham, Switzerland: Springer, 2018, pp. 194–208.

[2]. M. La Torre, J. Dumay, and M. A. Rea, ''Breaching intellectual capital: Critical reflections on big data security,'' Meditari Accountancy Res., vol. 26, no. 3, pp. 463–482, 2018.

[3]. S. W. Pritchard, G. P. Hancke, and A. M. Abu-Mahfouz, ''Cryptography methods for software-defined wireless sensor networks,''

in Proc. IEEE 27th Int. Symp. Ind. Electron. (ISIE), Jun. 2018, pp. 1257–1262.

[4]. G. Verma, M. Liao, D. Lu, W. He, X. Peng, and A. Sinha, ''An optical asymmetric encryption scheme with biometric keys,'' Opt. Lasers Eng., vol. 116, pp. 32–40, May 2019.

[5]. H. N. Almajed, A. S. Almogren, and A. Altameem, ''A resilient smart body sensor network through pyramid interconnection,'' IEEE Access, vol. 7, pp. 51039–51046, 2019.

[6]. R. Zuccherato, ''Elliptic curve cryptography support in entrust,'' Entrust Datacard, Ottawa, ON, Canada, Tech. Re. 1.0, May 2000.

[7]. M. Tyagi, M. Manoria, and B. Mishra, ''A framework for data storage security with efficient computing in cloud,'' in Proc. Int. Conf. Adv. Comput.Netw. Inform. Springer, 2019, pp. 109–116.

[8]. J. Louw, G. Niezen, T. D. Ramotsoela, and A. M. Abu-Mahfouz, ''A key distribution scheme using elliptic curve cryptography in wireless sensor networks,'' in Proc. IEEE 14th Int. Conf. Ind. Inform. (INDIN), Jul. 2016, pp. 1166–1170.

[9]. G. Kanda, A. O. Antwi, and K. Ryoo, ''Hardware architecture design of aes cryptosystem with 163-bit elliptic curve,'' in Advanced Multimedia and Ubiquitous Engineering. Singapore: Springer, 2018, pp. 423–429.

[10]. Z. E. Dawahdeh, S. N. Yaakob, and R. R. B. Othman, ''A new modification for menezes-vanstone elliptic curve cryptosystem,'' J. Theor. Appl. Inf. Technol., vol. 85, no. 3, p. 290, 2016.

[11]. L. Ferretti, M. Marchetti, and M. Colajanni, ''Fog-based secure communications for low-power IoT devices,'' ACM Trans. Internet Technol., vol. 19, no. 2, p. 27, 2019.

[12]. F. Albalas, M. Al-Soud, O. Almomani, and A. Almomani, ''Security-aware coap application layer protocol for the Internet of things using elliptic curve cryptography,'' Power (mw), vol. 15, no. 3A, p. 151, 2018.

[13]. S. Khan and R. Khan, ''Elgamal elliptic curve based secure communication architecture for microgrids,'' Energies, vol. 11, no. 4, p. 759, 2018.

[14]. S. Liu, H. Yao, and X. A. Wang, ''Fast elliptic curve scalar multiplication for resisting against spa,'' Int. J. Comput. Sci. Eng., vol. 17, no. 3, pp. 343–352, 2018.