ISSN: 2454-9940

# INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol19, Issue 1,2025

# ELLIPTIC ENCRYPTION-BASED EFFECTIVE ASYMMETRIC ENCRYPTION DESIGN

[1] NADUKUDA SRAVYA, [2] KOTHA KUMMARI SRIRAM, [3]YERRAGUNTA JAGADISHWAR, [4]ABHINAY POTHUGANTI, [5]GANGAM SUMITH REDDY, [6] Mr. POTHINENI VENKATESWARA RAO, [7]Mr. SAI SYAM,

[12345]Student Department of DS, Narsimha Reddy Engineering College, Maisammaguda (V), Kompally, Secunderabad, Telangana-500100.

[6]Assistant Professor ,Department of CSE, Narsimha Reddy Engineering College, Maisammaguda (V), Kompally, Secunderabad, Telangana-500100.

[7]Assistant Professor, Department of Mechanical Engineering, Narsimha Reddy Engineering College, Maisammaguda (V), Kompally, Secunderabad, Telangana-500100.

## Abstract—

There must be encryption for the smart grid's data transmissions since the data they include a great deal of sensitive information. At now, most encryption schemes use asymmetric encryption. An elliptic curve encryption technique with identity authentication is suggested as a solution to the problems with existing asymmetric encryption, which involves a lot of computation and storage space requirements and does not guarantee the integrity of encrypted information. As a first step, we present a nine-stage encrypted message transmission technique that uses elliptic curve encryption to decrease computation and storage consumption. Then, to guarantee the message's integrity and secrecy, we conduct security analysis and The strategy presented in this study is shown to be better by the experiment simulation.

Keywords—asymmetric encryption; authenticated encryption; elliptic curve encryption

## INTRODUCTION

Since the power system stores a considerable deal of sensitive information and network attacks are becoming more common as a result of IT advancements, protecting the power system is crucial. Currently, the distribution network's geographical spread is extensive. There are a plethora of approaches to heterogeneous network access. There are a lot of power terminals. The communication is in plaintext as well. A major security risk is created by these issues. It is essential to transmit data encrypted. Two broad classes characterize the state of the art in encryption algorithms: There are two types of encryption: symmetric and asymmetric. In symmetric encryption, the password is used for both the encryption and decryption processes. So, it can provide sufficient protection and safety when the encryption is sent to the participants in the chat [1]. The suggested algorithm mechanism's implementation entails creating a cipher for both the sender and the recipient, and then using that cipher to encrypt the sender's letter.

When transmitting messages using a cipher, the export end is responsible for receiving and sending the encrypted material. The client then used the previously generated cipher decryption information to decode the message. The advantages of this method are its straightforward process, quick realization, and little technical burden. However, there are significant drawbacks to the technique, such as the fact that it struggles with managing key exchange in the network and that, in the absence of digital signatures and certificates, it requires the user to find a trustworthy channel to get the cipher. A trustworthy encryption method is required due to the algorithmic flaws mentioned above. An asymmetric algorithm was developed for the purpose of solving problems. Every session user gets their own unique pair of keys: one for public cipher usage in encrypting communications and another for private use in decrypting them. Prior to the encryption process, the recipient acquires a program that allows them to get the sender's public key. The next step is for the client-sending device to encrypt the message using the received key. During decryption, the client's accepting equipment employs the private cipher.

This was said to be dependent on AES in reference [2]. The algorithm's distribution network communication encryption technology, however, this technique improves encryption efficiency by reducing the number of iterations. Security is significantly compromised by the AES algorithm because to its tremendous complexity. As a result of the method's massive power storage overhead, which necessitates an increase in the physical equipment of the key distribution center, reference[3] developed a power terminal authentication approach for lower computer abilities. A method for data signatures based on the RSA algorithm was proposed in Reference [4]. Under conditions of 1024 or greater modulus, it necessitates good security but significantly raises computational complexity. With a shorter key, Elliptic-Curve Cryptography (ECC) achieves the same degree of security as the RSA technique. Hence, Elliptic Curve-Cryptography is extensively used in low power devices that need to keep their performance. Few sources from the past that made use of the ECC system really explained how to do the value-to-message conversion. Not only do these programs provide several improvements in some areas, but they also complement ECC's own security advancements. For instance, the method of encoding and mapping the message to the chosen curve was not specified in References[5-7], which presented an ECC-based technique. These recommendations lessen the computing process and power usage in the problem. While an ECC-based picture encryption approach was suggested in References[8], increasing the cipher size results in more storage cost and encryption/decryption computations.

Information mapping and inverse mapping on ECC were suggested in References[9], however this approach is susceptible to selected plaintext steps that are used to encrypt ciphertext. There are significant implementation-level security vulnerabilities associated with the aforementioned ECC-based approaches for protecting nodes with low-capacity resources. Furthermore, due to ineffective identity verification, many techniques are unable to guarantee the security of encrypted communications. To ensure the privacy and security of all network communications, this article will provide an elliptic curve encryption method that makes use of authenticated encryption. Various encryption attempts can't break the system. The attacks include both plaintext and ciphertext. Additionally, the plan works well for devices that have little processing power and few other resources.

# II. ELLIPTIC CURVE CRYPTOGRAPHY

Cryptosystems based on elliptic curves originate from elliptic curves over discrete logarithmic fields. If p is greater than 3, then the elliptical curve p and the set of variables y are satisfied.

$$y^2 = x^3 + a \cdot x + b \bmod p \quad (1)$$
$$a, b \in \mathbb{Z}_p, 4 \cdot a^3 + 27 \cdot b^2 \neq 0 \bmod p$$

The absence of self-intersection and vertices in the network indicates that the elliptic curve is non-singular. You may see an example of an elliptic curve in Figure 1.

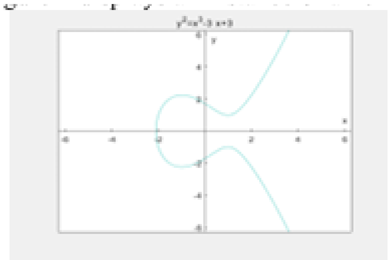


Fig. 1 Example of elliptic curve

The additive group operation's elliptic curve was defined as'+ '. Take P = (x1, y1) and Q = (x2, y2) as examples. Thus, P plus Q equals (x1, y1) plus (x2, y2), which equals (x3, y3). In certain cases, such as when P=Q, the process of adding P and Q equals (x1,y1) + (x1,y1) = 2P is known as point multiplication. The primary function of ECC is group multiplication[10, 13]. It is the number of cases when the points in a group double. The component d that makes up group multiplication is G = (xi, yi). The basis point on the elliptic curve is G=(xi, yi), and d is an integer that is termed the private key. Hence, the public key point is the output of the dG operation, which is a d-fold processing of G. In ECC, finding d at the moment of polymerization is impossible, even if the community cipher and principle point G are known. This is known as the ECDLP, or the elliptic curve discrete logarithm problem [14]. There are essentially four phases to ECC: Community cypher computation is the first step. One possible way to transmit the community cipher in ECC is to multiply the privately-owned cipher by the primary point, denoted as G [15–18]. Since ECC procedures include encrypted numbers, improving scalars in elliptic curves may lead to technology that can turn letters into numbers [19]. Avoiding cryptographic attacks (such as those using ciphertext or plaintext) requires That which is critical for data encoding. Third, the data must be signed to ensure that the sender is the intended recipient and to forestall tampering from other sources. Mapping the approved and encrypted messages to the elliptic curve is the last step [20]. As a crucial security measure, mapping messages to curves may merge the second and fourth steps of any proposed method to guarantee that the encrypted characteristics are unaltered. In addition, many techniques overlook the fact that signature encryption information is a crucial step. Signing the encrypted data ensures that it remains private and uncompromised. This converts it into an AE scheme, or authenticated encryption.

## III. SECURE AND EFFICIENT ENCRYPTION SCHEME

The nine-step process begins with the generation of system parameters and continues with the following steps: encoding the data and shining a light on it using an elliptic curve; the encryption mapping point; the signing of encrypted messages; the verification of the receiving data; the decryption of the data; the decoding and decoding of data; and finally, the conversion of decoded data into XMLText. Security message encoding, mapping, and ECC capabilities are offered by an AE. It should be noted that the first step is a crucial operation, and the proposed schemes fail to take into account the main stage—the system required to allow AE—by failing to recognize the need of using a community cipher to encrypt the message. Figure 2 displayed the nine steps of the proposed plan.

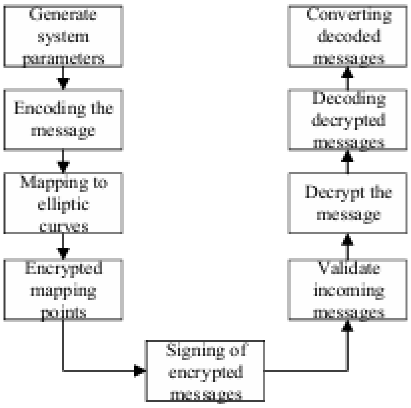
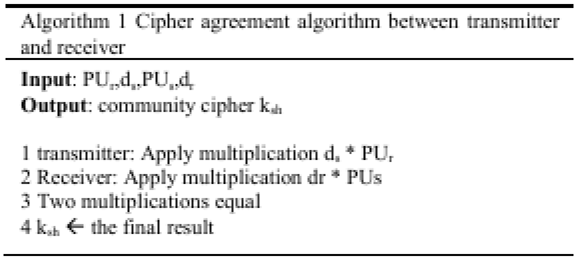

Generate system parameters
Encoding the message
Mapping to elliptic curves
Encrypted mapping points
Signing of encrypted messages
Validate incoming messages
Decrypt the message
Decoding decrypted messages
Converting decoded messages

Fig. 2 Flow chart of the suggested scheme

## System parameter generation

Creating a common cipher is the primary benefit of this stage. Making use of the encryption to secure the elliptic curve mapping points. For each session in this scenario, the symbols utilized by the system are explained in Table 1. To encrypt the elliptic curve mapping points, the sender must first build a shared session cipher, ksh. You can see the process of creating a cipher in Figure 3. Both the sender and the recipient may benefit from ECDLP by using it to unite the community cipher. To illustrate this procedure, use the following algorithm:



```
Algorithm 1 Cipher agreement algorithm between transmitter and receiver
Input: $PU_r$,$d_s$,$PU_s$,$d_r$
Output: community cipher $k_{sh}$

1 transmitter: Apply multiplication $d_s * PU_r$
2 Receiver: Apply multiplication dr * PUs
3 Two multiplications equal
4 $k_{sh}$ ← the final result
```

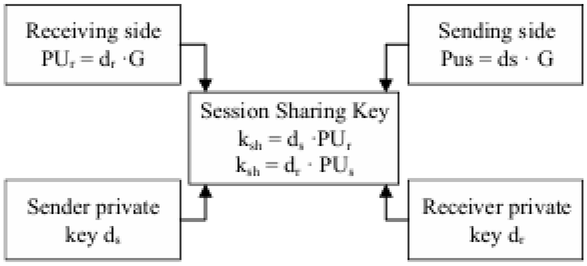
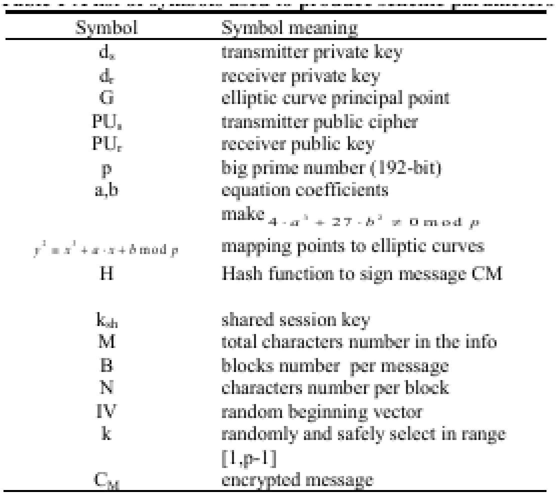


Fig. 3 Creating a shared key

Table 1 A list of symbols used to produce scheme parameters



| Symbol | Symbol meaning |
|---|---|
| $d_s$ | transmitter private key |
| $d_r$ | receiver private key |
| G | elliptic curve principal point |
| $PU_s$ | transmitter public cipher |
| $PU_r$ | receiver public key |
| p | big prime number (192-bit) |
| a,b | equation coefficients<br>make $4 \cdot a^3 + 27 \cdot b^2 \neq 0 \bmod p$ |
| $y^2 \equiv x^3 + a \cdot x + b \bmod p$ | mapping points to elliptic curves |
| H | Hash function to sign message CM |
| $k_{sh}$ | shared session key |
| M | total characters number in the info |
| B | blocks number per message |
| N | characters number per block |
| IV | random beginning vector |
| k | randomly and safely select in range [1,p-1] |
| $C_M$ | encrypted message |

## B. Encode messages as numeric values

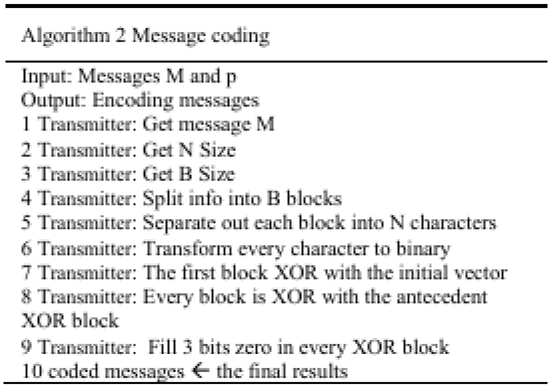
In order to overcome the security vulnerabilities that are associated with this method, the coding and mapping framework process described in Reference[29] is extended. Everything is broken down into its own block B. N characters are included in each block B. Here is the equation to calculate N:

$$N \leq \left\lfloor \frac{p-8}{8} \right\rfloor (2)$$

With p= 192 and N= 23, we get the following formula. In a similar vein, we may get the necessary block-B's number by dividing the total number of characters in M by n. For example, there are 44 blocks for texts that include 1000 characters. The following equation may be used to make it a reality:

$$B = \left\lceil \frac{M}{N} \right\rceil (3)$$

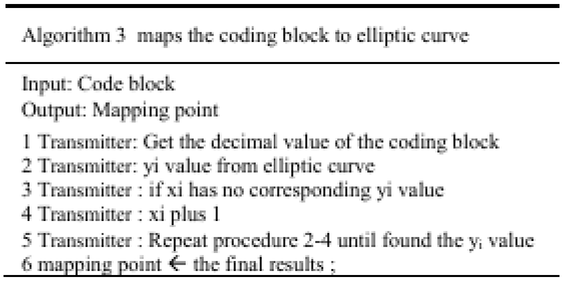
Encrypting each mapping point to the same length as p is the foundational premise of length separation. Also, before the mapping process, delete the first character from each block and fill it with the three bits that are zero. Following the acquisition of message M's block, the binary value of the characteristics of each group inside the block is assigned to each message. It begins with an initial vector and an XOR operation on the first one's binary value block. The subsequent blocks are then XORed with the antecedent block. The last step in mapping is to fill each XOR block with three bits. Prior to mapping to the elliptic curve, Algorithm 2 explains the message encoding technique.



```
Algorithm 2 Message coding
Input: Messages M and p
Output: Encoding messages
1 Transmitter: Get message M
2 Transmitter: Get N Size
3 Transmitter: Get B Size
4 Transmitter: Split info into B blocks
5 Transmitter: Separate out each block into N characters
6 Transmitter: Transform every character to binary
7 Transmitter: The first block XOR with the initial vector
8 Transmitter: Every block is XOR with the antecedent
XOR block
9 Transmitter:  Fill 3 bits zero in every XOR block
10 coded messages ← the final results
```

## Map message to elliptic curve

The elliptic curve is used to map the message, which implies that (xi, yi) indulges the curve in formula (1). Therefore, for each xi, it is necessary to get the matching value of yi. The encoded message's binary values are first translated to decimal. Then, to convert the value to EC, locate the matching yi value using the EC equation that was created. To map the message, use the steps outlined in Algorithm 3.



```
Algorithm 3  maps the coding block to elliptic curve
Input: Code block
Output: Mapping point
1 Transmitter: Get the decimal value of the coding block
2 Transmitter: yi value from elliptic curve
3 Transmitter : if xi has no corresponding yi value
4 Transmitter : xi plus 1
5 Transmitter : Repeat procedure 2-4 until found the $y_i$ value
6 mapping point ← the final results ;
```

## Encryption mapping points

A lot of schemes go over the encryption mapping point because they think the mapping step is enough to secure the message. However, this is not the case since elliptic curve mappings to points imply that ECDLP hardness may be obtained by multiplying these points by the privately held cipher. Adding each point to ksh encrypts them in this paper. Because of this, finding mapping sites without a common cipher is challenging. Figure 4 demonstrates the steps involved in the process of encrypting mapping points.

"data" means information in its plural form, not in its single form.

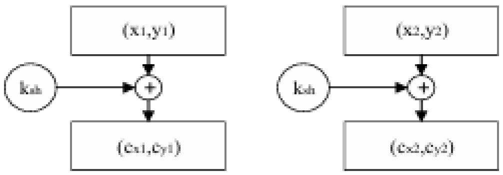
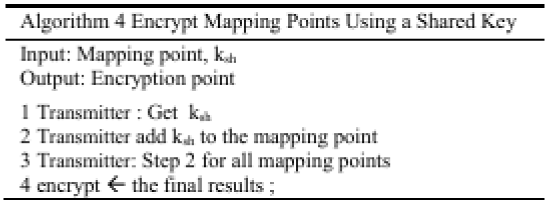


Fig.4 Encrypt mapping points using shared keys

The algorithm for encrypting mapping points using shared key ks :



```
Algorithm 4 Encrypt Mapping Points Using a Shared Key
Input: Mapping point, $k_{sh}$
Output: Encryption point
1 Transmitter : Get  $k_{sh}$
2 Transmitter add $k_{sh}$ to the mapping point
3 Transmitter: Step 2 for all mapping points
4 encrypt ← the final results ;
```

## Signature and Verification of Encrypted Messages

Secrecy and fullness of information transfer between the two parties are guaranteed by the AE approach in the scheme. We use the following approach to sign messages in order to implement the AE method: The elliptic curve is used to map all encryption points in every encrypted message CM. In addition, IV, kG, and CM are included in the delivered message Msent. Hence, the suggested approach employs ECDSA to Msent in order to produce authenticated encrypted communications. In Fig.5, we can see the signature procedure shown.

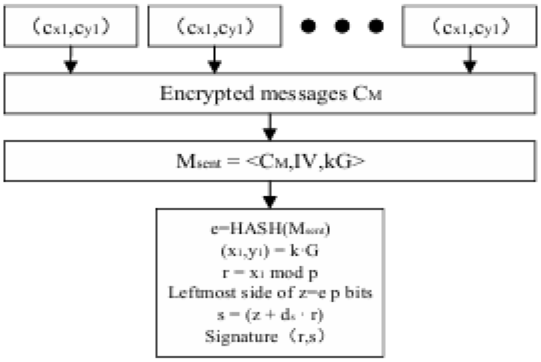
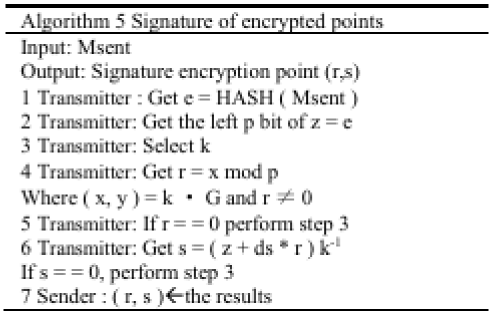


Fig. 5 Sign encrypted messages

Algorithm 5 describes how to sign CM.



```
Algorithm 5 Signature of encrypted points
Input: Msent
Output: Signature encryption point (r,s)
1 Transmitter : Get e = HASH ( Msent )
2 Transmitter: Get the left p bit of z = e
3 Transmitter: Select k
4 Transmitter: Get r = x mod p
Where ( x, y ) = k  •  G and r ≠ 0
5 Transmitter: If r = = 0 perform step 3
6 Transmitter: Get s = ( z + ds * r ) $k^{-1}$
If s = = 0, perform step 3
7 Sender : ( r, s )←the results
```

The signed message may be authenticated and decrypted by the recipient using the transmitter's public cipher. Figure 6 illustrates the procedure for receiver-involved validation.

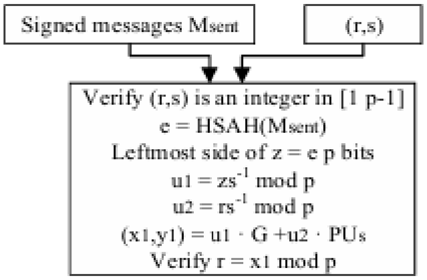

Signed messages Msent
(r,s)
Verify (r,s) is an integer in [1 p-1]
e = HSAH(Msent)
Leftmost side of z = e p bits
u1 = zs^-1 mod p
u2 = rs^-1 mod p
(x1,y1) = u1 · G +u2 · PUs
Verify r = x1 mod p

Fig. 6 Validation message

Substantiating and Confirming Secure Messages The steps that follow are in direct opposition to those that came before. In ECC, subtracting is the opposite of adding. This method additionally verifies that the encryption step is reversible. The receiver has to deduce the encrypted point using ksh before they can decode it. Figure 3 depicts the receiver's ksh production, whereas Figure 7 depicts the decryption of the encryption points.

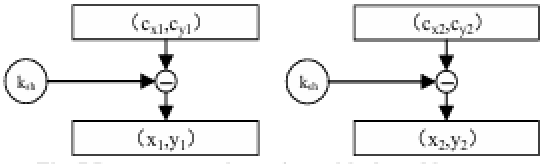


Fig. 7 Decrypt mapping points with shared keys

Similar to the encryption step, the decryption step obtains the mapping point by using the shared key ksh reverse operation.

```
Algorithm 6 Use a shared key to decrypt encryption points
Input: Encryption point, $k_{sh}$
Output: Mapping point
1 Recipient : Get $k_{sh}$
2 Recipient : Subtract $k_{sh}$ from the encryption point
3 Recipient : Perform step 2 on all encryption points
4 Mapping point ← the final results
```

## Decoding a decrypted message

A set of mapping points is the end product of the decryption phase. Two components, xi and yi, make these up. Points to elliptical curves are the only things that yi is used for. The only variables utilized in the decoding process are xi, which stand for the binary values used during the plain-text conversion. Algorithm 7's decoding step contained this method.

```
Algorithm 7 Decodes Mapping Points to Binary Values
Input : Mapping point
Output : Code block
1 Recipient: Get $x_i$ value from mapping point
2 Recipient: transform $x_i$ to binary value
3 Recipient: Removes 3 bits per block fill
4 Recipient: Initial vector XOR with first block
5 Recipient: Each block is XORed from the previous block
6 binary value ← the results
```

## Convert the decoded message into plaintext

Transforming binary data into useful characters is the final step. These represent the unencrypted message. For further information on how to transform binary values into plaintext, see M. Algorithm 8 below:

```
Algorithm 8 Converts binary values to plaintext
Input : Binary value
Output : plaintext message M
1 Input the encoded binary code into the receiver system
2 Decode all blocks binary code by receiver terminal
3 Output the decode result in form of plaintext message M
```

## SECURE AND EFFICIENT ENCRYPTION SCHEME

Any encryption technique must include security analysis. When we put our system up against other ECC schemes, the experimental findings reveal that ours is the best.

## Intercept assault

If an eavesdropper listens in on a communication session and steals the data sent during an encrypted message exchange, he may decipher the messages and give them back to the original sender. Furthermore, our plan effectively foils this assault: If an attacker were to listen in on the communication process and decipher the clear message, he would need to start a new session with the receiver parties in order to resend the information that has been falsified. Nevertheless, the end-communication-parties will reject the new session due to incorrect signature bits. This means that the initial vector of the recorded packet does not match the new session's initial vector.

## Attacks with extensibility

In a scalability attack, the decryption message is altered since the encoded message was fabricated. Any value may be XORed with the starting vector by the attacker. The following is the demonstration that the system presented in this work can withstand extension attacks:Interception of encrypted messages in transit between communication parties is possible. Because an outside hacker may decipher the transmission, piece together the false data, and resend it. However, the recipient will ignore the communication since the signature is useless.

## Modeling of performance

In this part, we will go over the experimental results that support our goal of proposing a very efficient AE encryption technique. 1) Each block's fill size was reduced. If there is no matching point on the elliptic curve for point xi while mapping it to the curve, then add 1 to xi and continue. We utilize a three-bit filling as the representation, whereas other systems use the ASCII table to encode the clear message into binary bits. They employed fixed eight-bits to represent each letter and mapped to the elliptic curve. While simulating secp192k1, NIST-224, and secp256k1 elliptic curves, we also produce 1000 random integers of 192 bits, 224 bits, and 256 bits, respectively, and assign them to the relevant elliptic curves in this article. It is possible to map about half of the values in the first round, and around three quarters in the second. And it's 88% in the third round. More than 98% of the data has been mapped into the elliptic curve in the sixth round of iterations.

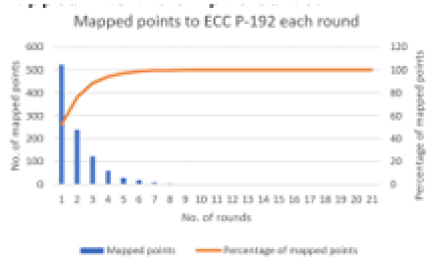


Fig. 8 Mapping random keys to elliptic curves secp192k1

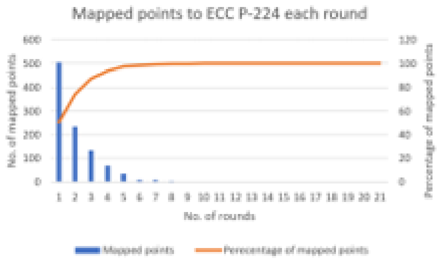


Fig. 9 Mapping random keys to elliptic curves NIST-224

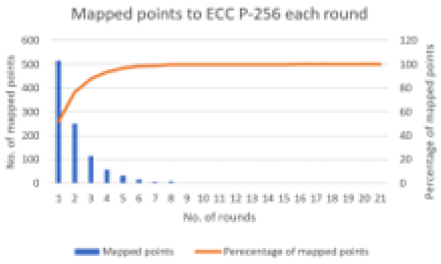


Fig. 10 Mapping random keys to elliptic curves secp256k1

Operations involving encoding and decoding Just after the last character of the last affiliation line, insert a hard return. After that, adhere the first affiliation copy. For each subsequent affiliation, repeat the process as needed.

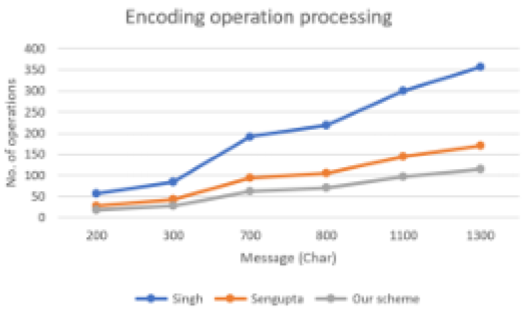


Fig. 11 The number of operands required to encode the message

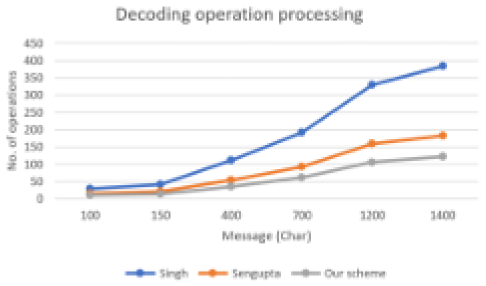


Fig. 12 The size of operands for decoding

## CONCLUTION

To efficiently encode and map messages to elliptic curves, this work presents an AE method that is based on these curves. It has been shown via simulation and analysis that the suggested method outperforms competing technologies with respect to attack resistance, filling size, and the amount of encoding and decoding operations.

## REFERENCES

*[1]. Z Chuan-fu, Y Jiang, S Wan-zhong, S Jin-hai. Cryptographic Key Agreement Protocol Simulation[C] // 2010 Sixth International Conference on Semantics, Knowledge and Grids. Beijing, 2010: 418-419.*

*[2]. WANG Chengshan, LI Peng, YU Hao. Development and characteristic analysis of flexibility in smart distribution network[J]. Automation of Electric Power Systems, 2018, 42(10): 13-21.*

*[3]. CAO Yijia, LI Qiang, TAN Yi, et al. A comprehensive review of Energy Internet: basic concept, operation and planning methods, and research prospects[J]. Journal of Modern Power Systems and Clean Energy, 2018, 6(3): 399-411.*

*[4]. Leng X , Chen G , Bai J , et al. General Design of Smart Grid Monitoring Operation Big Data Analysis System[J]. Dianli Xitong Zidonghua/Automation of Electric Power Systems, 2018, 42(12):160-166.*

*[5]. L. Ferretti, M. Marchetti, and M. Colajanni. Fog-based secure communi-cations for low-power IoT devices[J]. ACM Trans. Internet Technol., vol. 19,no. 2, p. 27, 2019.*

*[6]. Albalas F , Al-Soud M , Almomani O , et al. Security-aware CoAP Application Layer Protocol for the Internet of Things using Elliptic-Curve Cryptography[J]. International Arab Journal of Information Technology, 2018, 15(3a):550-558.*

*[7]. Sarmadullah K , Rafiullah K . Elgamal Elliptic Curve Based Secure Communication Architecture for Microgrids[J]. Energies, 2018, 11(4):759.*

*[8]. Singh L D , Singh K M . Image Encryption using Elliptic Curve Cryptography[J]. Procedia Computer Science, 2015, 54:472-481.*

*[9]. Sengupta A , Ray U K . Message mapping and reverse mapping in elliptic curve cryptosystem[J]. Security & Communication Networks, 2016.*

*[10]. Yin Y , Wu L , Peng Q , et al. A Novel SPA on ECC with Modular Subtraction[C] // 2018 12th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID). Xiamen, 2018: 179-182.*

*[11]. Galbraith S D , Frederik V . Computational problems in supersingular elliptic curve isogenies[J]. Quantum Information Processing, 2018, 17(10):265-.*

*[12]. Wu T , Wang R . Fast unified elliptic curve point multiplication for NIST prime curves on FPGAs[J]. Journal of Cryptographic Engineering, 2019.*

*[13]. Shahroodi T , Bayat-Sarmadi S , Mosanaei-Boorani H . Low Latency Double Point Multiplication Architecture Using Differential Addition Chain Over $GF(2^m)$[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2019:1-9.*

*[14]. Zhang J , Cui J , Zhong H , et al. PA-CRT: Chinese Remainder Theorem Based Conditional Privacy-preserving Authentication Scheme in Vehicular Ad-hoc Networks[J]. IEEE Transactions on Dependable and Secure Computing, 2019:1-1.*

*[15]. Z. Liu and H. Seo. IoT-NUMS: Evaluating NUMS Elliptic Curve Cryptography for IoT Platforms[J]. IEEE Transactions on Information Forensics and Security, 2019, 14(3):720-729.*

*[16]. Shah D P , Shah P G . Revisting of elliptical curve cryptography for securing Internet of Things (IOT)[C] // 2018 Advances in Science and Engineering Technology International Conferences (ASET). Dubai, Sharjah, Abu Dhabi, United Arab Emirates, 2018: 1-3.*

*[17]. Fournaris A P , Dimopoulos C , Moschos A , et al. Design and leakage assessment of side channel attack resistant binary edwards Elliptic Curve digital signature algorithm architectures[J]. Microprocessors and microsystems, 2019, 64(FEB.): 73-87.*

*[18]. A G Reddy, A K Das, V Odelu, A Ahmad, J S Shin. A privacypreserving three-factor authenticated key agreement protocol for client–server environment[J]. Journal of Ambient Intelligence and Humanized Computing, 2019, 10(2): 661–680.*

*[19]. Naji A K . Elliptic Curve Video Encryption in Mobile Phone Based on Multi-Keys and Chaotic Map[J]. Al-Mustansiriyah Journal of Science, 2018, 29(2): 106. [20] Reyad O . Text Message Encoding Based on Elliptic Curve Cryptography and a Mapping Methodology[J]. 2018.*