

Text and Image Encryption using ECC

Kaviarasu C¹, Kishore G², Nithish P³, Dr G A Sathish Kumar⁴

Students, Department of Electronics and Communication Engineering^{1,2,3}

Professor, Head of the Department, Department of Electronics and Communication Engineering⁴

Sri Venkateswara College of Engineering, Chennai, India

Abstract: *This project combines Elliptic Curve Cryptography (ECC) with Random Matrix Cryptography (RMC) for image encryption and ECC with the Lorenz algorithm to provide a comprehensive solution to data security. Chaotic equation for text encryption. While ECC-Lorenz uses chaotic dynamics for text encryption, ECC-RMC uses matrix operations to strengthen image encryption. The efficacy of both strategies in safeguarding the confidentiality and integrity of transmitted data is demonstrated through experimental validation, which therefore improves network communication security. A multi-layered encryption architecture is established by the integration of ECC-RMC and ECC-Lorenz, providing a strong defense against cyber attacks and unlawful access. This new encryption paradigm promises improved resilience in digital information transfer, marking a significant leap in data security measures. The suggested method gives a flexible way to combine computational complexity and mathematical rigor.*

Keywords: Elliptical curve cryptography(ECC),Random matrix cryptography(RMC),ECC-Lorenz Algorithm, Chaotic equations

I. INTRODUCTION

In today's digital landscape, where data breaches and cyber threats loom large, ensuring the confidentiality and integrity of sensitive information is paramount. Cryptography stands as a cornerstone in the defense against unauthorized access, providing robust encryption techniques to safeguard digital assets. Among these techniques, Elliptic Curve Cryptography (ECC) has emerged as a formidable solution, offering a unique combination of security, efficiency, and scalability. This paper explores the application of ECC in securing two fundamental types of digital assets: text and images. By harnessing the mathematical elegance of elliptic curves, ECC provides a robust foundation for cryptographic protocols that enable secure communication, storage, and transmission of information. Moreover, ECC offers several distinct advantages over traditional cryptographic systems, making it particularly well-suited for modern digital environments. One of the primary advantages of ECC lies in its ability to provide equivalent security with smaller key sizes compared to other public-key cryptosystems such as RSA. This results in reduced computational overhead and storage requirements, making ECC ideal for resource-constrained environments such as mobile devices and IoT (Internet of Things) devices. Additionally, ECC's mathematical properties contribute to faster encryption and decryption operations, facilitating efficient data processing without compromising security. Furthermore, ECC exhibits strong resistance to quantum computing attacks, ensuring long-term security against emerging threats. Unlike many traditional encryption schemes vulnerable to quantum algorithms, ECC's mathematical structure remains robust even in the face of quantum adversaries, offering a future-proof solution for data security. In this paper, we will delve into the theoretical foundations of ECC, elucidating its mathematical principles and highlighting its advantages over conventional cryptographic systems. Subsequently, we will explore practical implementations of ECC-based encryption for both text and image data, addressing key management, performance considerations, and real-world applications. Through a combination of theoretical insights and practical demonstrations, this paper aims to showcase the efficacy of ECC in enhancing data security across diverse digital domains. By leveraging ECC's unique advantages, practitioners and researchers can deploy robust encryption solutions to protect sensitive information, mitigating risks and fortifying the resilience of digital ecosystems against evolving threats.

II. PROBLEM DEFINITION

In contemporary information systems, ensuring the security and confidentiality of data and images during transmission and storage is paramount. However, existing encryption methods often encounter challenges that hinder their effectiveness in adequately protecting sensitive information. These challenges include

1. Cryptographic Vulnerabilities:

Traditional encryption techniques may be susceptible to cryptographic attacks, such as brute force attacks or cryptanalysis, which compromise the confidentiality of encrypted data.

2. Performance Limitations:

Some encryption methods may suffer from performance limitations, leading to delays in data transmission or processing, particularly when dealing with large datasets or high-resolution images.

3. Larger Key Sizes:

As the key size increases, cryptographic operations such as encryption, decryption, and key generation become more computationally intensive. This results in longer processing times and higher resource utilization, which can be impractical. Larger keys require more storage space to store and manage. This can be problematic, especially in embedded systems, mobile devices, and IoT devices where storage capacity may be limited.

III. THEORETICAL BACKGROUND

Automated testing utilizes advanced machinery and algorithms to swiftly and accurately assess PCBA quality, enhancing efficiency and precision. In contrast, manual testing relies on human intervention, which can be slower and prone to errors. While manual testing may offer meticulous scrutiny, it often lacks scalability and consistency. Automated methods, however, ensure uniformity in testing procedures, contributing to higher throughput and reliability in identifying defects, crucial for meeting stringent quality standards in PCBA manufacturing.

A . Fundamentals of ECC:

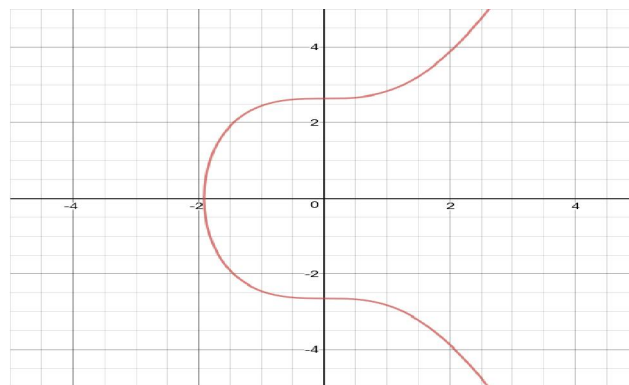


Figure 1: Elliptic curve

Elliptic Curves, defined by Equation (1), play a pivotal role in modern cryptography due to their security and efficiency advantages. The NSA Suite B has identified Elliptic Curve Cryptography (ECC) as a cornerstone of future cryptographic algorithms. ECC offers superior efficiency, requiring smaller key sizes compared to traditional methods like RSA. Elliptic Curves can be defined over various fields, including complex numbers (C) and finite fields modulo. Overall, ECC promises to revolutionize cryptographic practices by enabling more efficient and scalable encryption schemes.

$$y^2 = (x^3 + ax + b) \bmod p \rightarrow (1)$$

Time to break in MIPS years	RSA/DSA Key size	ECC key size	RSA/ECC key size ration
10^4	512	106	5:1
10^9	768	132	6:1
10^{11}	1024	160	7:1
10^{20}	2048	210	10:1
10^{79}	21000	600	35:1

Figure 2: Examination of the Identical Security Level for A few Normally utilized Cryptographic Key Sizes

B . Point Addition and Point Doubling:

In the context of Elliptic Curve Cryptography (ECC), the basic point addition operation involves combining two distinct points P and Q on the elliptic curve to compute a third point R. Given points $P(x_1, y_1)$ and $Q(x_2, y_2)$, with $P \neq 0$, $Q \neq 0$, and $P \neq -Q$, the resulting point $R(x_3, y_3)$ is computed using the elliptic curve equation. Specifically, the coordinates x_3 and y_3 of the new point R are determined by applying the point addition formula on the elliptic curve, ensuring that R lies on the same curve. This fundamental operation forms the basis for subsequent cryptographic processes, such as point multiplication, within the ECC framework.

$$\begin{aligned} X_3 &= M^2 - x_1 - x_2 \pmod{p} \\ Y_3 &= M^*(x_1 - x_3) - y_1 \pmod{p} \end{aligned} \rightarrow (2)$$

$$M = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{IF } P \neq Q \\ \frac{3(x_1)^2 + a}{2y_1}, & \text{IF } P = Q \end{cases} \rightarrow (3)$$

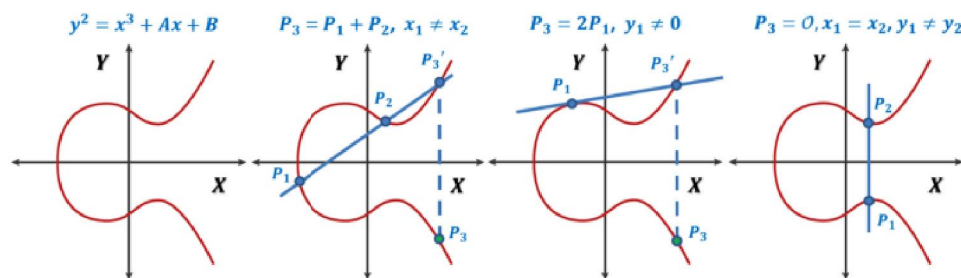


Figure 3: Point addition and Point doubling

C . Point Multiplication:

Point multiplication in Elliptic Curve Cryptography (ECC) involves a combination of point addition and point doubling operations, denoted as $k \times P = Q$. Point addition combines two points J and K to produce a new point L i.e., $L = J + K$, while point doubling involves adding a point J to itself to obtain L (e.g., $L = 2 \times J$). For example, given an integer scalar k, it is multiplied with a point P to generate a new point Q on the Elliptic Curve (EC). The process involves iteratively applying point addition and point doubling, commonly referred to as the 'doubling and adding' method to derive Q.

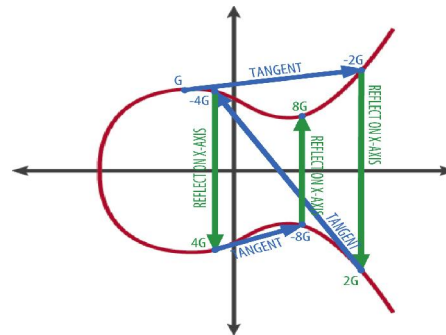


Figure 4: Point Multiplication

D . Computing Modular Inverses in Elliptic Curve Cryptography (ECC):

The `modInv(x, n)` function plays a critical role in Elliptic Curve Cryptography (ECC) by computing the multiplicative inverse of x modulo n . Initially, it verifies whether x and n are coprime, ensuring the existence of a valid inverse. If not coprime, x is adjusted accordingly. Utilizing the Extended Euclidean Algorithm, it computes the greatest common divisor (d) of x and n , along with coefficients (a) that satisfy Bézout's identity ($ax + ny = d$). These coefficients are then reduced modulo n to derive the multiplicative inverse ($xInv$) of x modulo n . This calculated inverse is indispensable in ECC for various operations, including key generation and modular arithmetic, thereby upholding the security and integrity of cryptographic processes.

E . Securing Communication with Lorenz Chaotic Encryption:

The Lorenz system falls under the broader umbrella of chaos theory, which is a branch of mathematics and physics that deals with complex, unpredictable behavior in deterministic nonlinear dynamical systems

Deterministic Systems:

Chaos theory deals with deterministic systems, meaning that their future behavior is completely determined by their initial conditions and the equations governing their evolution. Despite this determinism, chaotic systems can exhibit highly irregular and seemingly random behavior.

Sensitivity to Initial Conditions:

One of the defining characteristics of chaotic systems, including the Lorenz system, is sensitivity to initial conditions, often referred to as the butterfly effect. This means that tiny differences in the starting conditions can lead to vastly different outcomes over time. As a result, long-term predictions become inherently uncertain beyond a certain time horizon.

Strange Attractors:

Chaotic systems often exhibit strange attractors, which are geometric objects in phase space that represent the system's long-term behavior. Unlike simple attractors such as fixed points or limit cycles, strange attractors have a fractal structure and are characterized by complex, non-repeating patterns. The Lorenz attractor is a famous example of a strange attractor associated with the Lorenz system.

Periodic Orbits and Chaos:

In many chaotic systems, including the Lorenz system, there exists a range of parameter values for which the system exhibits chaotic behavior. However, outside this range, the system may exhibit periodic behavior, such as stable fixed points, limit cycles, or other regular patterns. The transition from regular to chaotic behavior often occurs through a process known as a bifurcation.

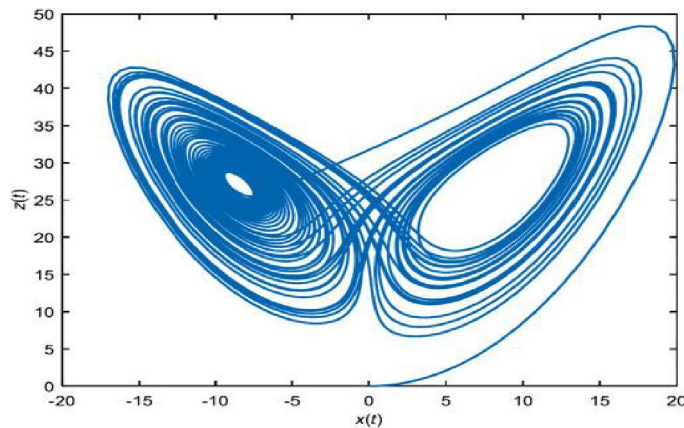


Figure 5: Lorenz Chaotic System

Universality:

Chaotic behavior often exhibits certain universal properties that are independent of the specific details of the system. For example, the Feigenbaum constants describe the universal scaling behavior of bifurcations near the onset of chaos in one-dimensional maps, regardless of the particular form of the map.

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y - x), \\ \frac{dy}{dt} &= x(\rho - z) - y, \quad \rightarrow (4) \\ \frac{dz}{dt} &= xy - \beta z.\end{aligned}$$

where,

x, y, and z are the state variables representing the current state of the system.

σ , ρ , and β are parameters that control the behavior of the system.

t represents time.

F . Random Matrix Cryptography:

In the encryption process, generating a random matrix is a crucial step. Here, a 256x256 matrix is created using the public key as the seed values so that the same random matrix is generated on both the sender and receiver side without ever transmitting the whole random matrix, with each element filled with random values between 0 and 1. To ensure reproducibility while maintaining randomness, a pseudorandom number generator is employed, initialized with a seed value. This seed value acts as the starting point for the random number generation process, enabling consistent results across different executions. By incorporating this random matrix, the encryption scheme gains a vital layer of unpredictability, bolstering the security of the encrypted image. Through element-wise multiplication with this matrix, the encryption process becomes more robust, ensuring the confidentiality of sensitive information during transmission.

Random Matrix Cryptography (RMC):

Random Matrix Cryptography (RMC) is a cryptographic method that utilizes randomly generated matrices to encrypt and secure data, particularly in the context of image encryption and secure communication systems. The fundamental idea behind RMC is to leverage the complexity of matrix operations to enhance the security of encrypted data. Here's a concise explanation of Random Matrix Cryptography and its key concepts

Random Matrix Generation:

In RMC, encryption keys are represented as random matrices. These matrices are generated using a secure random number generator, ensuring that they possess desirable cryptographic properties such as uniformity and unpredictability.

Resistance to Cryptanalytic Attacks:

RMC offers robust security against various cryptanalytic attacks due to the complexity introduced by matrix operations. The randomness and size of the encryption matrix contribute to the strength of the cryptographic scheme, making it challenging for attackers to reverse-engineer the plaintext from the ciphertext without knowledge of the correct key.

IV. IMPLEMENTATION

ECC Diffie-Hellman:

Diffie-Hellman (DH) is a cryptographic technique that enables two parties to share a secret key via an insecure channel. The key exchange is based on modular exponentiation and depends on the discrete logarithm problem's difficulty to ensure security. Elliptic Curve Cryptography (ECC) is a public-key cryptography that utilizes the algebraic structure of elliptic curves over finite fields. ECC provides comparable security to classic public-key cryptosystems (such as RSA), but with significantly smaller key sizes, making it more efficient in terms of computation and bandwidth. Unlike regular Diffie-Hellman, ECC Diffie-Hellman exchanges keys using elliptic curve cryptography rather than modular arithmetic. ECDH offers the same security guarantees as traditional DH but with smaller key sizes and thus faster computation and less bandwidth overhead.

Key Exchange Process:

In ECC Diffie-Hellman, both parties agree on a common elliptic curve and a base point on that curve. Each party generates a private key, which is a randomly chosen integer, and computes its public key by multiplying the base point by its private key (using scalar multiplication). The parties then exchange their public keys. Finally, each party combines its own private key with the received public key using scalar multiplication to compute the shared secret key. Overall, ECC Diffie-Hellman provides a secure method for two parties to establish a shared secret key over an insecure channel.

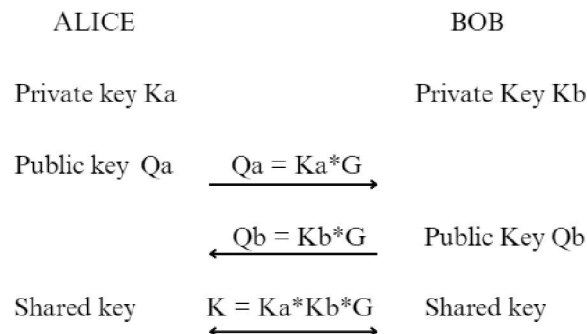


Figure 6: Diffie-Hellman key exchange

TEXT ENCRYPTION AND DECRYPTION:

Public Key Generation Process:

Elliptic Curve Parameters:

This section sets up the parameters for elliptic curve cryptography (ECC), a powerful cryptographic technique based on the algebraic structure of elliptic curves. The chosen parameters include a prime number p , and coefficients a and b defining the elliptic curve equation.

$$y^2 = (x^3 + ax + b) \bmod p. \rightarrow (5)$$

Check Point on Curve:

A function is defined to ensure that any given point lies on the elliptic curve. This is crucial for ECC operations, as all calculations must be performed on points that satisfy the curve equation.

Generate Private Key:

In ECC, each party involved generates a private key, which is a randomly selected integer within a specific range. Here, the code generates a private key within the range of $[1, p-1]$.

Select Base Point:

ECC operations typically involve a base point G on the elliptic curve. In this code, an arbitrary starting point $(x,y) = (5,1)$ is chosen as the base point. It's essential to verify that this point lies on the curve.

Generate Public Key:

The public key is derived from the private key and the base point using scalar multiplication. The process involves adding the base point to itself multiple times (as dictated by the private key) to obtain the public key. This public key is used by other parties to encrypt messages for the receiver.

ENCRYPTION:

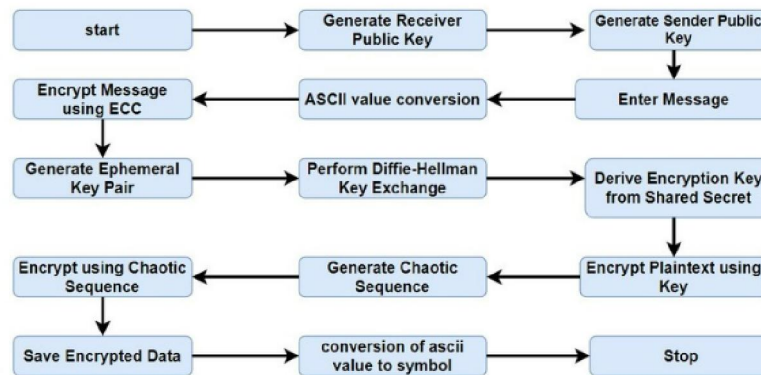


Figure 7: Flow Diagram of ECC text encryption

The sender initiates the encryption process for the message they wish to send. Encryption is essential to secure the message during transmission, ensuring that only authorized parties can access its contents.

Elliptic Curve Cryptography (ECC):

The message is encrypted using ECC, a robust and efficient encryption method based on the computational difficulty of solving discrete logarithm problems on elliptic curves. This involves generating an ephemeral key pair, performing a Diffie-Hellman key exchange, deriving an encryption key from the shared secret, and finally encrypting the message using this key.

Lorenz Chaotic Map:

Chaotic sequences, such as those generated by the Lorenz chaotic map, are often employed in encryption schemes to enhance security. These sequences exhibit sensitive dependence on initial conditions, making them unpredictable and suitable for cryptographic applications.

Encryption using Chaotic Sequence:

The chaotic sequence obtained from the Lorenz chaotic map is used to encrypt the message. This adds an additional layer of security to the encryption process, as the chaotic sequence serves as a random key stream for encrypting the message

DECRYPTION USING ECC:

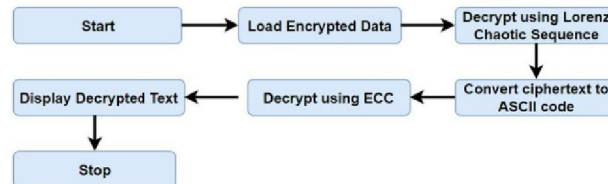


Figure 8: Flow Diagram of ECC text Decryption

Load Data: The receiver loads the encrypted message and the necessary parameters from a saved file. This ensures that they have access to all the information required for decryption.

Text Decryption:

The receiver initiates the decryption process to recover the original message sent by the sender. Decryption is the reverse process of encryption and is essential for retrieving the original plaintext from the encrypted ciphertext.

Lorenz Chaotic Model:

The receiver generates the chaotic sequence using the same Lorenz chaotic map used by the sender. This ensures that the receiver's chaotic sequence matches the one used during encryption, allowing for successful decryption.

Elliptic Curve Cryptography (ECC):

The encrypted message is decrypted using ECC. This involves deriving the shared secret from the receiver's private key and the sender's ephemeral public key, deriving the encryption key from the shared secret, and decrypting the message using this key.

IMAGE ENCRYPTION AND DECRYPTION:

ENCRYPTION USING ECC:

Elliptic Curve Parameters:

a, b, and p are parameters that are fundamental to defining the elliptic curve used in the encryption process, a and b are coefficients, and p is a prime number.

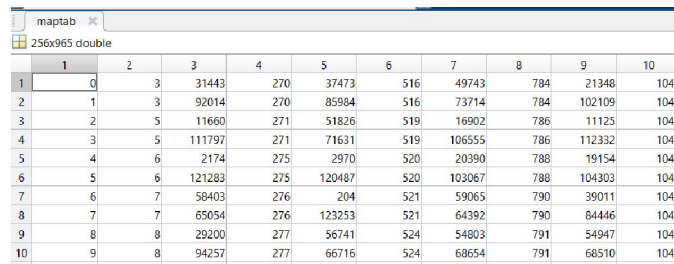
a = 5379,

b = 2438,

p = 123457

Mapping Table:

The maptab variable is loaded. This table correlates pixel intensities to points on the elliptic curve, facilitating the encryption process. The table was generated by fixing the curve constants a, b, and p, and collecting all the points by varying the x and y values from 1 to p and ensuring that the points exist on the curve by using mathematical equations and doing a consistent mapping scheme across encryption and decryption.



	1	2	3	4	5	6	7	8	9	10
1	0		31443	270	37473	516	49743	784	21348	1041
2	1		92014	270	85984	516	73714	784	102109	1041
3	2		11660	271	51826	519	16902	786	11125	1043
4	3		111797	271	71631	519	106555	786	112332	1043
5	4		2174	275	2970	520	20390	788	19154	1044
6	5		121283	275	120487	520	103067	788	104303	1044
7	6		58403	276	204	521	59065	790	39011	1046
8	7		65054	276	123253	521	64392	790	84446	1046
9	8		29200	277	56741	524	54803	791	54947	1049
10	9		94257	277	66716	524	68654	791	68510	1049

Figure 10: maptab matrix

User Input:

The user provides a sender private key (K) and a receiver private key (Nb). These keys serve as essential parameters in the encryption process, contributing to the uniqueness and security of the encryption.

Base Point Selection:

A base point (G) is selected on the elliptic curve. The chosen point is critical as it forms the basis for ECC operations during encryption.

Public Key Generation:

The public key (Pb) of the receiver is computed by multiplying the base point (G) with the private key (Nb). This step generates a key that can be shared with the sender for secure communication.

Encryption Key Generation:

An encryption key (EK) is derived by multiplying the public key (Pb) with the random key (K). This key is used to perform ECC operations on pixel intensities during encryption.

Image Encryption:

Each pixel of the input image is encrypted using ECC and the mapping table. The intensity of each pixel is mapped to corresponding points on the elliptic curve, and encryption involves point addition and mapping to obtain new pixel intensities. The encrypted pixel values are stored in the Enc_Msg variable for further processing or transmission.

ENCRYPTION USING RMC:

ECC Public Key Integration:

The ECC (Elliptic Curve Cryptography) public key, previously generated, serves as a critical element in enhancing the security of the encryption process. By leveraging the values comprising the ECC public key, a unique seed for the random number generator (RNG) is derived. The seed generation process involves aggregating the values of the ECC public key, ensuring that variations in the public key result in distinct RNG seeds.

Random Matrix Generation:

With the RNG seed set based on the ECC public key, a random matrix (RM) of specified dimensions (256x256) is generated. The random matrix serves as the foundation for creating cryptographic keys and introducing randomness into the encryption process.

Random Matrix Part Generation:

The random matrix (RM) is divided into four distinct parts, facilitating the encryption of different segments of the image. Each part is individually stored as an image file, enabling traceability and verification of the encryption process.

Sorting and Key Generation:

The values within each part of the random matrix are sorted in ascending order, generating unique sequences of indices. These sorted indices act as cryptographic keys, ensuring the confidentiality and integrity of the encrypted image segments.

Image Partitioning and Encryption:

The original image is partitioned into four segments, mirroring the division applied to the random matrix. Utilizing the sorted indices obtained from the random matrix, each image segment undergoes encryption, employing a substitution cipher approach to enhance security.

Visualization of Encrypted Image Parts:

Encrypted versions of the image segments are displayed, providing visual confirmation of the encryption process and highlighting the transformation applied to the image data.

Final Encrypted Image Generation:

The encrypted image is constructed by combining the encrypted segments, resulting in a composite ciphertext representing the original image's encrypted form. This final encrypted image serves as the secure output ready for transmission or storage.

DECRYPTION USING RMC:**Partial Decryption with RNG Seed:**

To initiate the decryption process, the RNG seed derived from the ECC public key is utilized to partially decrypt the encrypted image. This partial decryption step involves applying a scalar multiplication operation between the encrypted image and the random matrix, yielding a partially decrypted image.

Encrypted Image Part Generation for Decryption:

The partially decrypted image is divided into four segments, aligning with the segmentation applied during encryption. Each segment contains encrypted data awaiting decryption using the corresponding sorted indices derived from the random matrix.

Sorting and Decryption with Cryptographic Keys:

Leveraging the sorted indices obtained during encryption, each segment of the partially decrypted image undergoes decryption. The decryption process reverses the encryption transformation, utilizing the sorted indices to rearrange the encrypted data and unveil the original image content.

Visualization of Decrypted Image Part:

Decrypted versions of the image segments are visualized, demonstrating the successful reversal of the encryption process and confirming the accuracy of the decryption outcome.

Final Decrypted Image Generation:

The decrypted image is reconstructed by combining the decrypted segments, effectively restoring the original image content and completing the decryption process.

DECRYPTION USING ECC:

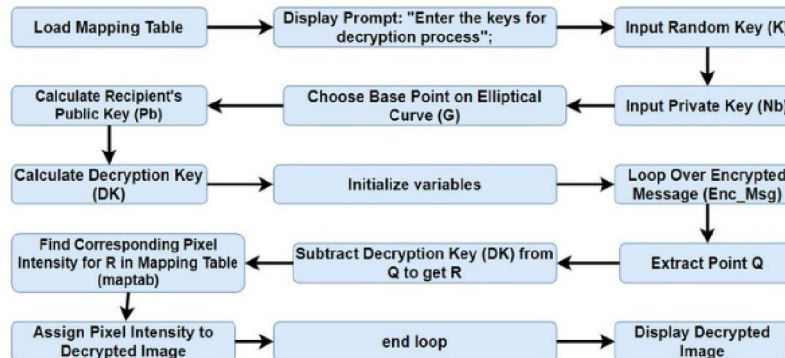


Figure 11: Flow Diagram of image decryption using ECC

Base Point Selection:

Similar to encryption, a base point (G) is selected on the elliptic curve.

Public Key Generation:

The public key (P_b) of the receiver is computed by multiplying the base point (G) with the private key (N_b).

Decryption Key Generation:

A decryption key (DK) is derived by multiplying the public key (P_b) with the random key (K). This key is used to perform ECC operations during decryption.

Image Decryption:

Each encrypted pixel of the message is decrypted using ECC and the *mapping table*. Decryption involves point subtraction and mapping to obtain original pixel intensities. The decrypted pixel values are stored in the decrypt variable.

Image Enhancement:

Optionally, the decrypted image undergoes enhancement, such as histogram equalization, to improve visual quality. This step enhances the contrast and details of the decrypted image, making it easier to analyze.

VI. RESULTS

TEXT ENCRYPTION AND DECRYPTION:

```

Command Window
>encrypt
Text Encryption in Sender Side
Enter the message: 'hello world!!'
Initial Cipher Text using ECC:
110 107 114 114 117 38 125 117 120 114 106 39 39
Final Cipher Text using Lorenz:
nh/coDR<00010
fx >>
  
```

Figure 12: Text encryption

```

Command Window

Final Cipher Text using Lorenz:
nh{coK<008i0
>> TY_Main
Text Decryption in Receiver Side
Decrypted Text:
hello world!!
Decrypted Message is: hello world!!
***** Completed *****
fx >>
  
```

Figure 13: Text decryption

IMAGE ENCRYPTION AND DECRYPTION:

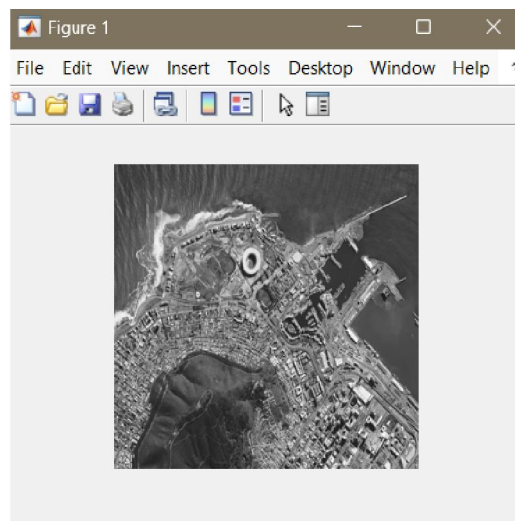


Figure 14: Uploaded Image

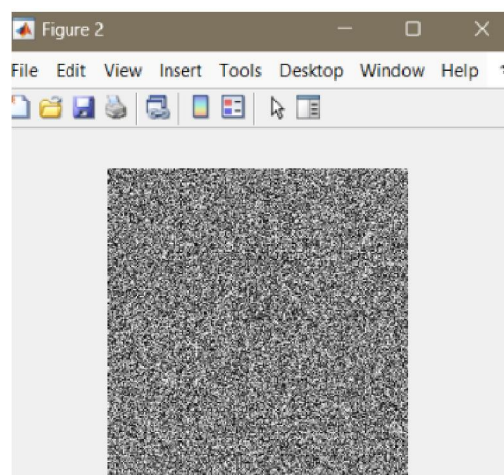


Figure 15: Encrypted Image

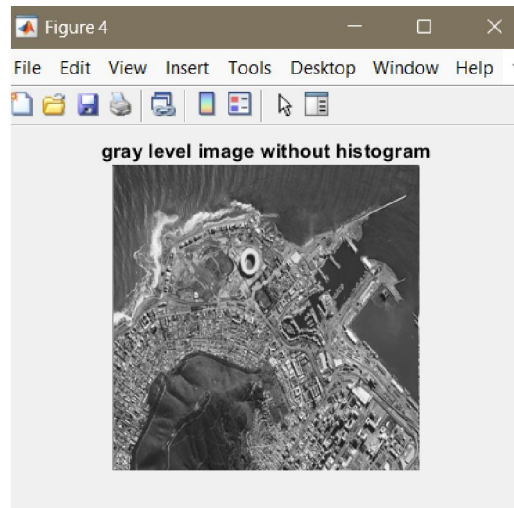


Figure 16: Decrypted Image

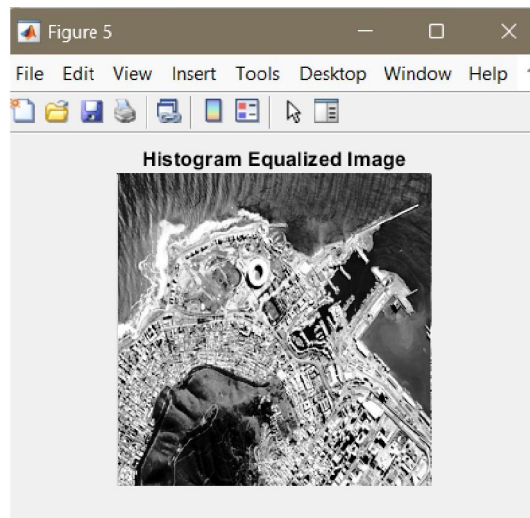


Figure 17: Enhanced decrypted Image

PSNR:

The PSNR block computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed, or reconstructed image.

$$PSNR = 10 \log_{10} \left(\frac{R^2}{MSE} \right) \rightarrow (6)$$

MSE:

The mean-square error (MSE) and the peak signal-to-noise ratio (PSNR) are used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower the error.

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N} \rightarrow (7)$$

Normalized Cross Correlation:

We compute the NCC value between the origins and encrypted images to measure the similarity of corresponding pixel locations. The aim of this step is to utilize the underlying geometric cue since the entire lane markings lie on the road plane and all the road points are mapped into the same global coordinates.

$$NCC = \frac{\sum_{(i,j) \in W} [f_1(i,j) - f_1] [f_2(i,j) - f_2]}{\sqrt{\sum_{(i,j) \in W} [f_1(i,j) - f_1]^2 \sum_{(i,j) \in W} [f_2(i,j) - f_2]^2}} \rightarrow (8)$$

SI.NO	IMAGES	PSNR(dB)	MS	NCC
1	Satellite_1.jpg	26.678	119.68	0.99663
2	Satellite_2.png	23.024	324.11	0.98737
3	Lena.jpg	27.039	127.56	0.99725
4	Man.jpg	23.847	268.18	0.99429
5	Monkey.jpg	31.184	49.513	0.99871

Figure 18: Results of Statistical experiment values

VII . CONCLUSION

Our project marks a significant milestone in the realm of secure communication and data protection, as we've successfully implemented an integrated system for encrypting satellite images and text messages using Elliptic Curve Cryptography (ECC). By harnessing the cryptographic power of ECC, our system offers a robust encryption framework capable of safeguarding sensitive data transmissions against unauthorized access and malicious attacks. The seamless integration of ECC with chaos-based encryption techniques further enhances the cryptographic strength and confidentiality of our system. Through meticulous testing and validation, our implementation has demonstrated its efficacy in ensuring data integrity and privacy across various communication channels and data types.

VIII. FUTURE SCOPE

Looking ahead, several promising avenues for future research emerge in the field of ECC and cryptographic systems. One compelling direction is the exploration of post-quantum ECC algorithms capable of withstanding the cryptographic threats posed by quantum computing. Additionally, research efforts could focus on advancing ECC-based authentication protocols and key management schemes for secure multi-party communication and identity verification. Furthermore, investigating the integration of ECC with emerging technologies such as blockchain, Internet of Things (IoT), and edge computing could unlock new opportunities for secure and decentralized applications in various domains, including finance, healthcare, and smart cities. By continuously pushing the boundaries of cryptographic innovation, we aim to address the evolving security challenges of the digital age and empower individuals and organizations to safeguard their data with confidence and resilience.

IX. APPLICATIONS

1. Secure Messaging and Email Encryption:

- **End-to-end Encryption:** ECC can be used to encrypt text messages or emails to ensure that only the intended recipient can decrypt and read the content.
- **Digital Signatures:** ECC can be used for message authentication and integrity verification, ensuring that the received message has not been tampered with and originates from the expected sender.

2. Secure File Transfer:

- **Encryption of Image Files:** ECC can encrypt image files, ensuring that only authorized users with the correct decryption keys can access the images.
- **File Integrity:** ECC can verify the integrity of transferred image files, guaranteeing that the received file matches the original and has not been modified during transit.

3. Secure Storage and Cloud Services:

- **Client-Side Encryption:** ECC can be used to encrypt files (including images) before uploading them to cloud storage services. This ensures that even if the cloud provider is compromised, the stored data remains confidential.
- **Access Control:** ECC can be utilized to manage access control to encrypted files, allowing only authorized users to decrypt and access specific content.

4. Digital Rights Management (DRM):

- **Protecting Intellectual Property:** ECC can be used to encrypt copyrighted images, preventing unauthorized users from accessing or distributing them without proper authorization.

5. IoT (Internet of Things) Security:

- **Secure Device Communication:** ECC can secure communications between IoT devices, ensuring that data transmitted between devices remains confidential and cannot be intercepted or altered by unauthorized parties

X. ACKNOWLEDGMENT

We would like to express our deepest gratitude to Sri Venkateswara College of Engineering and our mentor **Dr. G A Sathish Kumar** (Professor, Head of the department, Department of ECE) for providing us with the resources and necessary support to complete this project. We are also extremely grateful for their invaluable guidance, feedback, and encouragement throughout this process. Their expertise and insight were instrumental in shaping the direction of this project and refining its execution. Thank you for sharing your time, knowledge, and expertise with us. Your contributions have been immeasurable and will always be remembered

REFERENCES

- [1]. 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) "Design and Implementation of Satellite Image Encryption by Using ECC"
<https://ieeexplore.ieee.org/xpl/conhome/8977133/proceeding>
- [2]. A. K. Najj. (2018) "Elliptic curve video encryption in mobile phone based on multi-keys and chaotic map." Al-Mustansiriyah Journal of Science 29
https://www.researchgate.net/publication/329025363_Elliptic_Curve_Video_Encryption_in_Mobile_Phone_Based_on_Multi-Keys_and_Chaotic_Map
- [3]. Z.E. Dawahdeh, S. N. Yaakob and R.R. B. Othman. (2016) "A new modification for Menezes-Vanstone Elliptic Curve Cryptosystem." Journal of Theoretical and Applied Information Technology 85
https://www.researchgate.net/publication/318021056_A_New_Image_Encryption_Technique_Combining_Elliptic_Curve_Cryptosystem_with_Hill_Cipher
- [4]. L. D. Singh. and K. M. Singh. (2015) "Implementation of text encryption using elliptic curve cryptography." Procedia Computer Science 54:73-82.
https://www.researchgate.net/publication/283186188_Implementation_of_Text_Encryption_using_Elliptic_Curve_Cryptography

- [5]. N. Koblitz. (1987) "Elliptic curve cryptosystems." Math. Comp. 48: 203-209.
<https://link.springer.com/article/10.1007/BF00203817>
- [6]. V. S. Miller. (1986) "Use of Elliptic Curves in Cryptography." In: Williams H.C. (eds) Advances in Cryptology — CRYPTO '85 Proceedings. CRYPTO 1985. Lecture Notes in Computer Science 218: 417-426
https://www.researchgate.net/publication/221355411_Use_of_Elliptic_Curves_in_Cryptography
- [7]. A. Shamir, L. Adleman, and R.L. Rivest. (1978) "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." Communications of the ACM 21 (2):120-126
<https://www.scirp.org/reference/referencespapers?referenceid=2170254>
- [8]. W. Diffie and M. E. Hellman. (1976) "New direction in cryptography." IEEE Transactions on Information Theory 22 (6): 644-654.
<https://www.sciencedirect.com/science/article/pii/0022000084900709>