Palestine Journal of Mathematics

Vol 14(3)(2025) , 429–450

Enhanced ECC-driven Text Encryption Scheme using Chaotic Maps and Rhotrices

S. Gupta, R. Narang, M. K. Patel, K. Gupta and K. Kumar*

Communicated by Manoj Patel

*Corresponding author

MSC 2010 Classification 12E20; 94A60.

Keywords and phrases: Chaotic Sequences, Peano Masks, Elliptic Curve Cryptography, Encryption, Information Security.

The authors would like to thank the reviewers and editor for their constructive comments and valuable suggestions that improved the quality of our paper.

Abstract This research paper presents an advanced encryption scheme that integrates chaotic maps, Peano masks, matrices, and Elliptic Curve Cryptography (ECC) to enhance data security through multiple layers of transformation and permutation. The proposed scheme leverages the logistic map to generate chaotic sequences, which are further refined using elliptic curve points to ensure uniqueness for each data block. Peano masks introduce additional complexity by permuting matrix positions based on a secret key, while ECC transformations introduce non-linearity and complexity to the encryption process. Further, authentication is added with the help of digital signature. Statistical tests, including the frequency test, block frequency test, cumulative sums test, runs test, and the longest run of one's test, further validate the robustness and security of the scheme. The effectiveness of the proposed scheme is demonstrated by its encryption and decryption times. The overall resilience of the proposed scheme against known and chosen plaintext/ciphertext attacks, combined with a large key space and high key sensitivity, underscores its strength and effectiveness. The detailed analysis of the encryption and decryption processes showcases the scheme's ability to provide secure communication through intricate mathematical and chaotic systems, ensuring that data integrity and confidentiality are maintained.

1 Introduction

In cryptography, various techniques are employed to ensure secure communication between the sender and receiver, limiting message visibility to authorized parties only. Significant advancements in elliptic curve cryptography (ECC) have led to its widespread use in encryption, decryption, digital signatures, authentication, and key agreement [1, 2, 3, 4, 5, 6, 7, 8] . This literature review thoroughly explores different encryption models, discussing their strengths, weaknesses, and potential applications. Amounas and El Kinani [9] proposed a permutation scheme using a non-singular matrix to counter such attacks, encrypting the mapped message with ECC. However, their method was susceptible to chosen plaintext attacks if the matrix used for multiplication was uncovered. Hankerson et al. [10] presented a method that maps each character of a message to points on an elliptic curve by multiplying the generator point G with the ASCII value of the character. Nevertheless, this approach was prone to frequency analysis attacks due to the repetition of encrypted characters in the ciphertext.

King [11] introduced a deterministic approach that interprets binary strings to map messages onto elliptic curves, ensuring that each message element is securely translated into a subgroup point. Paragas et al. [12] developed a Hill cipher chain algorithm based on modified Hill cipher

principles, incorporating XOR and shift operations for block encryption along with RADIX 64 encoding. Muthukuru and Sathyanarayana [13] proposed a mapping technique that utilizes XOR operations to map fixed-length strings to elliptic curve points; however, this method was vulnerable to man-in-the-middle and chosen plaintext attacks. Bh et al. [14] applied the Koblitz technique to probabilistically map messages onto an elliptic curve, but this approach was prone to collision attacks. Almajed et al. [15] highlighted the importance of padding bits in ECC mapping, examining optimal sizes to enhance both performance and security. Kumar et al. [16] introduced a method for secure and authenticated communication using ECC over finite fields, where plaintext characters associated with elliptic curve points were encrypted with shared keys. Keerthi and Surendiran [17] introduced a method where the plaintext message was first converted into ASCII values, then transformed into hexadecimal format and grouped according to the input size. The order of these hexadecimal values was then reversed, and the final ciphertext was generated through scalar multiplication on an elliptic curve. Kamalakannan and Tamilselvan [18] developed an encryption technique in which each character of the plaintext was converted to its ASCII value, mapped to specific affine points on an elliptic curve, and then modified using a matrix mapping scheme before being secured with ElGamal encryption. Naji et al. [19] proposed an encryption process that involved converting each plaintext character into a 16-bit decimal representation, segmenting these into larger integer blocks, and then encrypting them with a shared key. This approach replaced characters with affine points, eliminating the need for costly mapping techniques.

Tiwari and Kim [20] proposed an innovative method inspired by DNA-based ECC, where DNA genome sequences were used to assign values to different character sets within a message. This mapping utilized randomized character assignments based on pseudo-random data, playing a crucial role in securing encrypted messages. However, the approach did not fully address the encoding process that converts the message into numerical values for mapping to DNA sequences, which could make it vulnerable to certain encryption attacks. Nadeem et al. [21] introduced an efficient algorithm designed to safeguard data from unauthorized access and misuse, particularly in cloud computing environments. Their method involved converting plaintext into ASCII byte values, generating keys using a Non-Deterministic Bit Generator (NRBG), XNORing the keys with plaintext bits, and applying a matrix cipher encryption algorithm. This approach ensured that unique keys derived from the plaintext were used for decryption, thereby enhancing data security at various stages to generate the ciphertext.

Sattar et al. [22] proposed a novel text encryption scheme that combines elliptic curve cryptography with max-plus algebra-based wavelet transform, aimed at enhancing both security and efficiency in data protection. Almajed and Almogren [23] developed an authenticated encoding and mapping scheme using ECC metrics, which proved resilient against attacks and achieved optimized performance through careful selection of padding sizes and efficient encoding and decoding operations. However, their implementation of Cipher Block Chaining (CBC) mode restricted parallel processing across multiple processors or cores. Cahyono et al. [24] created encryption and decryption algorithms utilizing Type IVa MP-Wavelet analysis and synthesis techniques. Their encryption strategy involves ensuring that the total number of channels across all levels exceeds the number of characters in the plaintext, thereby improving security against brute-force attacks. The decryption process uses this encryption key along with a sequence derived from the binary encoding of detail components, further enhancing resistance to cryptographic attacks. This algorithm is noted for its efficiency, with a linear complexity relative to the number of characters in the plaintext. Zhao et al. [25] introduced the Lightweight Reputation-Based Consensus Mechanism (LRBCM) designed to secure Industrial IoT (IIoT) data. LRBCM employs node reputation values to simplify ledger consensus, thereby reducing communication overhead and system complexity. Sharma et al. [26, 27] proposed the TEXCEL encryption scheme for text and an image encryption scheme that maps characters to points on an elliptic curve. Gupta et al. [28] examined Business-to-Business (B2B) methods in healthcare, addressing the challenges of managing large volumes of smart device data. Their study focused on enhancing data transmission, patient access, and care quality while addressing security issues and suggesting techniques to protect sensitive healthcare information. Fu et al. [29] aimed to bolster security in Hyperledger Fabric by substituting the Elliptic Curve Digital Signature Algorithm (ECDSA) with the SM2 algorithm. Their method improves signature processes, reduces time complexity, and boosts overall system performance by incorporating SM2 into Fabric's Blockchain Crypto-

graphic Service Provider (BCCSP) module.

Sengupta and Ray [30] examined various message mapping schemes in elliptic curve cryptography, addressing their vulnerabilities to cryptanalysis and offering guidelines for effective and secure message mapping. They introduced a novel scheme aimed at resisting frequency analysis and other cryptanalytic attacks. Singh and Singh [31] developed a technique that maps message characters to affine points on an elliptic curve by grouping ASCII values and using the Big Integer function to generate these mapped points. Genç and Afacan [32] proposed an innovative approach that converts each character in a message to its hexadecimal Unicode representation, which differs from ASCII by covering a broader range of characters and using hexadecimal values ranging from one to six digits.

Azhar et al. [33] proposed a novel encryption scheme that combines the Pell sequence with elliptic curves. The process begins with scrambling plaintext through cyclic shifts, followed by obscuring the text using the Pell sequence, a weight function, and binary encoding. Additional confusion is introduced through permutations on elliptic curves, enhancing resilience against key sensitivity and statistical attacks. Murtaza et al. [34] developed a scheme for creating a fast and secure S-box generator tailored for lightweight cryptography, using small elliptic curves and binary sequences for text encryption. Das and Giri [35] proposed two new algorithms for transforming input messages into elliptic curve points, which helps reduce communication and computational costs while improving performance, especially for managing large text inputs.

Kordov [36] introduced a cryptographic system that utilizes a pseudorandom generator derived from two chaotic maps, showing resilience against various attacks through extensive cryptographic analysis. Ullah et al. [37] developed efficient S-box and pseudo-random number generators based on ordered Mordell elliptic curves, which demonstrated the ability to produce numerous distinct and cryptographically robust S-boxes and random sequences with minimal time and space complexity. Ataş and Güler [38] proposed an encryption-decryption method using the fractional-order Rössler chaotic system for securing image, sound, and text data, enhancing security through Master-Slave synchronization of the chaotic system. Murillo [39] applied the logistic map to create a symmetric encryption algorithm for text. Rihartanto et al. [40] designed a text encryption scheme that incorporates bit-based cube rotation to achieve confusion and diffusion, resulting in ciphertext with a high avalanche effect and low correlation coefficient, ensuring effective encryption regardless of the original content. Ivanova et al. [41] developed a text encryption algorithm using two Clifford attractors to improve message transfer security, which was validated through rigorous security testing.

In the age of digital communication and ubiquitous data exchange, the security of information has become paramount. Traditional cryptographic methods face increasing challenges due to the rapid advancement of computational capabilities and emerging cryptographic attacks. To address these challenges, novel encryption schemes that incorporate multiple layers of security and complexity are essential. This paper introduces an advanced encryption scheme that combines chaotic maps, Peano masks, matrices, and ECC to provide robust security for sensitive data.

In the realm of digital security, encryption remains a fundamental technique to safeguard sensitive information from unauthorized access. The evolution of cryptographic techniques has continually sought to address the challenges posed by increasingly sophisticated attacks and computational advancements. Modern encryption schemes leverage various mathematical and algorithmic innovations to enhance security and resilience. Among these innovations, chaotic systems play a significant role. Chaotic systems are characterized by their sensitivity to initial conditions and non-linear dynamics, which lead to pseudo-random behavior. These systems, such as the logistic map, are defined by iterative equations that produce sequences with properties similar to true random numbers. The chaotic nature of this map ensures that even slight variations in the initial condition led to significantly different sequences, which is beneficial for generating unpredictable cryptographic keys. ECC is another crucial advancement in modern encryption. ECC is a public-key cryptographic approach based on the algebraic structure of elliptic curves over finite fields. ECC offers comparable security to traditional methods like RSA but with shorter key lengths, making it computationally efficient. ECC's security is derived from the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP), which is computationally challenging.

Peano masks are used to permute data in encryption schemes. They are based on the Peano curve, a space-filling curve that covers a multidimensional space completely. By shuffling data posi-

tions according to a secret key, Peano masks enhance the diffusion and confusion properties of the encryption process. This permutation introduces additional complexity, making it harder for attackers to analyze patterns in the encrypted data. Matrices are used to organize and transform data blocks in encryption schemes. They provide a method for block-wise processing, allowing for structured modifications to plaintext data. Matrices undergoes a series of transformations, including chaotic substitutions and permutations. This technique enhances data obfuscation by applying non-linear and linear operations to each block, contributing to the overall security of the encryption scheme.

In our proposed encryption scheme, these concepts are integrated to create a robust cryptographic framework. Chaotic sequences generated from the logistic map introduce high entropy and randomness into the encryption process. ECC adds a layer of mathematical complexity through elliptic curve transformations, while Peano masks and matrices further obscure the data. This combination aims to improve security, resist attacks, and ensure that the encrypted data remains confidential and secure. The effectiveness of the proposed scheme is evaluated through various cryptographic tests, including entropy analysis, frequency tests, and randomness tests, to demonstrate its capability to produce high-quality encryption. By leveraging these advanced concepts, the scheme offers enhanced protection against contemporary cyber threats and potential vulnerabilities in traditional encryption methods.

The design of the proposed encryption scheme takes into account several critical gaps as given below:

(i) Chaotic sequences are often underutilized in encryption algorithms despite their ability to generate highly unpredictable sequences. Traditional methods may not fully leverage the potential of chaotic systems. The scheme uses chaotic sequences generated from the logistic map to ensure dynamic and unpredictable initial conditions for each data block. By incorporating the x-coordinate of an elliptic curve point and modifying it for each block, the scheme enhances the randomness and uniqueness of the chaotic sequence.

(ii) Standard data permutation techniques may not be sufficiently robust or secure, potentially making encrypted data vulnerable to attacks that exploit predictable patterns. The integration of Peano masks provides a sophisticated data permutation method. By shuffling matrix indices based on a secret key, the scheme introduces a high degree of data shuffling and complexity, making it more challenging for attackers to discern patterns.

(iii) Some encryption schemes may lack sufficient non-linearity, which is crucial for resisting linear and differential cryptanalysis attacks. The use of ECC introduces non-linearity through scalar multiplication. This non-linearity adds an additional layer of cryptographic strength and complexity, enhancing the scheme's overall security.

(iv) Ensuring high encryption quality and distinguishing ciphertext from plaintext can be challenging, particularly in maintaining statistical randomness and avoiding patterns. The scheme's effectiveness is evaluated through various metrics, including encryption quality, entropy analysis, and statistical tests. By demonstrating high EQ and robust randomness, the scheme addresses concerns related to the quality of encryption and ensures that the ciphertext is substantially different from the plaintext.

1.1 Objectives of the present work

The primary objectives of this paper are as follows:

(i) Develop a multi-layered encryption Scheme: To design and implement a novel encryption scheme that integrates chaotic maps, Peano masks, matrices, and ECC to enhance data security. The scheme aims to leverage the unique properties of each component to create a robust encryption method with increased security and complexity.

(ii) Utilize chaotic maps for initial conditions: To utilize the chaotic logistic map for generating pseudo-random sequences that serve as initial conditions for each block of data. The approach involves using the x-coordinate of an elliptic curve point, modified with block indices, to ensure unique and deterministic chaotic sequences for each data block.

(iii) Incorporate Peano masks for data permutation: To implement Peano masks to permute data within $n \times n$ matrix blocks. This permutation is guided by a secret key and aims to thoroughly shuffle data elements, enhancing the complexity of the encryption process and providing additional layers of security.

(iv) Apply ECC for non-linearity: To apply ECC to transform data blocks through rhotrix multiplication and rhotrix addition, introducing non-linearity and additional cryptographic strength. The ECC transformation is designed to be computationally efficient while providing robust security against potential attacks.

(v) Evaluate encryption quality and security: To assess the performance of the proposed encryption scheme through various metrics, including encryption quality, entropy analysis, and statistical randomness tests, and to demonstrate the scheme's resilience against common cryptographic attacks. Through these objectives, the paper aims to contribute to the field of cryptography by offering a sophisticated and practical encryption solution that enhances data security while addressing contemporary challenges in digital communication.

1.2 Organisation of the present work

The rest of the work is organized as follows: Section 2 illustrates the preliminaries necessary for understanding the paper. Section 3 highlights the proposed methodology. Section 4 provides architecture and work. The experimentation, results, and performance analysis of the proposed scheme are presented in Section 5. Practical implementation and proof of the proposed encryption scheme is provided in Section 6. Finally, sections 7 and 8 respectively present a discussion and conclude the paper with future scope.

2 Preliminaries

This section provides a foundational understanding of the key concepts, techniques, and tools utilized in the proposed encryption scheme.

2.1 Chaotic Maps

Chaotic maps are mathematical functions that exhibit sensitive dependence on initial conditions, producing complex and seemingly random sequences. These sequences are highly sensitive to small changes in the initial state, making them valuable for cryptographic applications. In the proposed scheme, the Logistic map is employed, defined by:

$$x_{n+1} = rx_n(1 - x_n),$$

where x_n is the current value in the sequence, r is the control parameter, typically within the range $[3.5, 4]$ to ensure chaotic behavior, and x_{n+1} is the next value in the sequence. The chaotic sequence generated is normalized to the byte range $[0, 255]$ to facilitate its use in data encryption.

2.2 Peano Curve

A Peano curve is a continuous curve that fills a two-dimensional space. It is defined recursively as follows:

(i) Base case:
For $n = 0$, the curve is just a single point.

(ii) Recursive case:
For $n = 0$, the curve is constructed from the Peano curve of the previous order $(n - 1)$ by dividing the space into nine smaller squares and applying a specific permutation of the previous curve in these squares.
The Peano curve is defined such that every point in a two-dimensional square can be reached by moving along the curve, which effectively "fills" the square.

2.3 Peano Mask

A Peano mask is a permutation matrix that results from applying the Peano curve to the data in a grid or matrix. Here's how you can derive it:

(i) Initialize a Grid: Start with an $N \times N$ grid, where $N = 2^n$ for some integer n.

(ii) Generate the Peano Curve:
Step 1: Construct the Peano curve for a base case $n = 0$.
Step 2: Use recursive subdivision to extend the curve to higher orders. The recursive steps involve subdividing each segment of the previous order into smaller segments and permuting these according to a predefined rule.

(iii) Create the Mask: The mask is derived by mapping the positions in the grid according to the order in which they are visited by the Peano curve. For a given matrix, the Peano mask is a permutation matrix that specifies how the data elements in the matrix should be rearranged based on the Peano curve.

2.4 Elliptic Curve Cryptography

ECC introduced in 1986 by Victor Miller and Neil Koblitz, emerged as a viable alternative to traditional public key cryptosystems like RSA and ElGamal. Mathematically intricate, ECC offers robust security and superior performance compared to its counterparts.
The general form of an elliptic curve E is expressed as: $y \equiv x^3 + ax + b(\mathrm{mod} p)$.
Here, a and b represents elements from a specific set (such as rational numbers, real numbers, integers modulo p, etc.), while x and y belong to the finite field $GF(p)$. In elliptic curves over prime fields $GF(p)$ with $p \geq 3$, the parameters a and b must satisfy the condition $4a^3 + 27b^2 \not\equiv 0(\mathrm{mod}\ p)$. Points on an elliptic curve are typically represented using Affine coordinates. Here are the key operations for working with points on an elliptic curve:

(i) Negation of a point: The negative of a point P is its reflection in the $x-$axis, denoted as $-P = (x, -y)$

(ii) Addition of two points: Given two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ on an elliptic curve, the sum of these points is calculated as:

$$P_3(x_3, y_3) = P_1(x_1, y_1) + P_2(x_2, y_2), \text{where}$$

$$x_3 = \lambda^2 - x_1 - x_2,$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

and

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}.$$

(iii) Doubling a point $P(x_1, y_1)$, is given by $P(x_2, y_2)$, where

$$x_2 = \lambda^2 - 2x_1,$$

$$y_2 = \lambda(x_1 - x_2) - y_1$$

and

$$\lambda = \frac{3{x_1}^2 + a}{2y_1}.$$

2.5 Elliptic Curve Diffie Hellman Key Exchange

Let m and n be the private keys of Alice and Bob respectively. Alice's public key is mG and Bob's public key is nG. These public keys are exchanged over an open channel. Alice then multiplies her private key with Bob's public key, and Bob multiplies his private key with Alice's public key. Both Alice and Bob will obtain the same result. This method of key exchange between users is known as the Elliptic Curve Diffie-Hellman (ECDH) key exchange.

2.6 Rhotrix [42]

Ajibade defined a 3–dimensional rhotrix, which is in some way between 2×2 and 3×3 matrices as follows:

$$R_3 = \left\langle \begin{array}{ccc} & a & \\ b & h(R) & d \\ & e & \end{array} \right\rangle,$$

where $a,b,d,e \in \mathbb{R}$, denote elements of a rhotrix and $h(R)$ represents the heart of an odd-dimensional rhotrix. This concept was further developed by Mohammed et al. [43] to extend the rhotrix structure to order n.

$$P_n = \left\langle \begin{array}{ccccccccc} & & & & a_1 & & & & \\ & & & a_2 & a_3 & a_4 & & & \\ & & a_5 & a_6 & a_7 & a_8 & a_9 & & \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ a_{\frac{t-n+2}{2}} & \cdots & \cdots & \cdots & a_{\frac{t+1}{2}} & \cdots & \cdots & \cdots & a_{\frac{t+n}{2}} \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & a_{t-8} & a_{t-7} & a_{t-6} & a_{t-5} & a_{t-4} & & \\ & & & a_{t-3} & a_{t-2} & a_{t-1} & & & \\ & & & & a_t & & & & \end{array} \right\rangle,$$

where $t = \frac{(n^2+1)}{2}$, $a_i \in \mathbb{R}$. In this context $h(R) = a_{\frac{t+1}{2}}$, serves as the heart of R_n. Generally, a rhotrix is denoted as $R = \langle h(R), a_i \rangle, 1 \le i \le t, i \ne \frac{t+1}{2}$. Heart oriented rhotrices are well-known in rhotrix literature. The algebra and analysis of rhotrices was defined by Ajibade. The addition and heart based multiplication of rhotrices is defined as
If $P = \langle h(P),\ a_i \rangle$ and $Q = \langle h(Q), b_i \rangle$ be two rhotrices, then

$$P+Q = \langle h(P)+h(Q),\ \ a_i+b_i \rangle,$$

$$PQ = \langle h(P)\ h(Q)\ ,\ \ a_i\, h(Q)\ +\ b_i\, h(P)\ \rangle,\ 1 \le i \le t,\ \ i \ne \frac{t+1}{2}.$$

Sani [44] proposed the alternative multiplication of rhotrices, row column multiplication which is similar to multiplication of two matrices.

$$PQ = \left\langle \begin{array}{ccc} & af+dg & \\ bf+eg & h(P)h(Q) & aj+dk \\ & bj+ek & \end{array} \right\rangle.$$

2.7 Finite Field [45]

A non-empty finite set F is defined as a finite field if it functions as an abelian group under both addition and multiplication. In computing, positive integers are typically represented as $n-$bit words, where n can be values such as $8, 16, 32, 64$, and so on. As a result, the maximum range of integers is $2^n - 1$. A Galois field, denoted as $GF(2^n)$, represents a finite field with 2^n elements, while $GF(p)$ includes the set $\mathbb{Z}_p$, where p is the largest prime number less than 2^n. The elements within this field are composed of $n-$bit words, as discussed in the works of Hun and Yen [46] and Hell and Johnson [47].

3 Proposed Methodology

The proposed encryption scheme integrates chaotic maps, Peano masks, matrix transformations, and ECC to enhance the security of data transmission. By using the $x-$coordinate of a shared elliptic curve point as the base initial condition for generating chaotic sequences, the scheme ensures that both the sender and receiver can independently produce the same sequence. Each

plaintext block is transformed through a series of operations: adding a chaotic sequence, applying a Peano mask permutation, and performing an ECC transformation. This multi-layered approach ensures robustness against various cryptographic attacks, providing a secure and efficient method for encrypting and decrypting data.

3.1 System model

Suppose Alice and Bob are two communicating parties and they want to share some message between them. Both, Alice and Bob agree on a common elliptic curve with generator point G.

(i) Alice chooses a random number t in such a way that it lies between $[1, l-1]$, where l is the order of generator point G. Alice keeps n_a as her private key.

(ii) Alice computes her public key as $P_a = n_a G$.

(iii) Bob selects a large random number n_b in such a way that it lies between $[1, l-1]$. He keeps n_b as his private key.

(iv) Public key of Bob is $P_b = n_b G$.

(v) Alice calculates her shared key, $A_S = n_a P_b = (p, q)$.

(vi) Bob calculates his shared key, $B_S = n_b P_a = (p, q)$.

(vii) Alice selects a random secret key k, this secret parameter will be sent to Bob with the help of digital signature.

3.2 Components of the Scheme

Chaotic Maps

Chaotic maps generate pseudo-random sequences that are used to introduce variability into the encryption process. We use the logistic map defined by:

$$x_{n+1} = r x_n (1 - x_n),$$

where x_n is the current value, r is the control parameter (we take $r \in (3.57, 4]$ for chaotic behavior, where p is the $x-$coordinate of shared key) and x_{n+1} is the next value in the sequence. The sequence is normalized to the byte range $[0, 255]$ for encryption purposes.

Initial Condition Determination:

Use a shared point $A_S = n_a P_b = (p, q)$ on the elliptic curve. The $x-$coordinate p serves as the base initial condition. To transform this $x-$coordinate into an initial condition suitable for chaotic systems, it is normalized to a value within the range of the chaotic system. Specifically convert the $x-$coordinate p to a value between 0 and 1. This is done by using the fractional part of p directly. The normalized value becomes the starting point for the chaotic sequence. This normalized value is crucial because chaotic systems require an initial condition in a specific range to produce pseudo-random sequences.

Both parties start with $x_0 = 1000 + 0.p$, that is, fractional part of $x-$coordinate of A_S is used to get a chaotic sequence with desirable number of terms.

Block Index Adjustment:

To ensure that each block of data in the encryption process uses a unique initial condition, the base initial condition (derived from the $x-$coordinate) is modified by appending or incorporating the block index. This adjustment ensures that each data block is encrypted with a different chaotic sequence, maintaining both security and uniqueness. Normalize the sequence to the byte range $[0, 255]$ for use in the encryption process.

Thus, the $x-$coordinate of a shared elliptic curve point is used to generate the base initial condition for a chaotic sequence. By normalizing this coordinate and adjusting it for each data block, a secure and unique chaotic sequence is created for encryption. This approach leverages elliptic curve properties to ensure that both parties in the communication process can deterministically generate and synchronize the chaotic sequences used in the encryption scheme.

Peano Masks

A Peano mask is created by shuffling the indices of a matrix. This mask will be used to reorder the elements of the matrix during encryption.

(i) Generating the Peano Mask:
The Peano mask is generated using a shared secret key k. This key ensures that both the sender and the receiver can generate the same mask for encryption and decryption.

(ii) Initialization:
The block size for our encryption is defined. For this example, we use a 4×4 block size, which means each block will contain 16 elements. Convert the secret key k into a seed for the random number generator to ensure reproducibility.

(iii) Peano Curve Generation:
A Peano curve is a space-filling curve that traverses a $2D$ grid in a continuous manner. To create a Peano mask, we generate a Peano curve that covers all positions in the 4×4 grid. The positions are indexed sequentially based on the order they are visited by the Peano curve.

(iv) Permutation:
The Peano mask is essentially a permutation of the indices of the 4×4 grid, which is derived by traversing the grid according to the Peano curve. The generated permutation is then used to shuffle the elements of each block during encryption.

- Row: The row index is found by performing integer division of the flat index i by the number of columns n.
- Column: The column index is found by calculating the remainder when the flat index i is divided by the number of columns n.

By using these formulas, we can convert any index from a $1-$dimensional list into its corresponding (row, column) position in a $2-$dimensional matrix.

(v) Applying the Peano Mask: During encryption, the elements of each 4×4 block of plaintext are reordered according to the Peano mask. During decryption, the inverse of the Peano mask is applied to restore the original order of the elements.

Matrices

Matrices are used to organize and transform data blocks.

(i) Block Formation:
Divide plaintext into 4×4 blocks. Pad the data if it doesn't fit perfectly into $16-$byte blocks.

(ii) Transformations:
Each block undergoes chaotic substitution by adding a portion of the chaotic sequence to each element of the block. Further, permute the block using the Peano mask.

Construction of Rhotrices

Here, we construct three rhotrices for encryption process.

(i) Construction of Message Rhotrix R_m :
Message rhotrix R_m is formed by arranging the above received blocks sequentially. The first five elements in this sequence become the elements of the first message rhotrix. The next five elements are used for the second message rhotrix, and this pattern continues. If the last rhotrix has fewer than five elements, it will be padded to complete it.

(ii) Construction of Chaotic Rhotrix R_c :
The chaotic rhotrix R_c is generated by taking the first five elements of a chaotic sequence to form the first chaotic rhotrix. For the second message rhotrix, the next five elements of the chaotic sequence are used to create the second chaotic rhotrix, and this process continues in the same manner. This method ensures the robustness of the encryption.

(iii) Secret Key Rhotrix R_s :
To construct the secret key rhotrix R_s, start by taking the $96-$bit shared key k and dividing it into six blocks. For the first element of the secret key rhotrix, XOR the first block with the second block; for the second element, XOR the second block with the third block; for the third element, XOR the third block with the fourth block, and so on. This process continues until all elements of the rhotrix are derived.

Elliptic Curve Cryptography

ECC is used to apply mathematical transformations to the data. In our scheme:

(i) ECC Parameters:
Define elliptic curve parameters, including a curve over a finite field and corresponding public/private keys.

(ii) ECC Transformation:
Use a simple rhotrix multiplication and rhotrix addition to transform each byte of the block. The ECC transformation is applied as follows:

$$C_1 \equiv R_m R_c + R_s (\text{mod } p)$$

This transformation introduces non-linearity and complexity.

3.3 Encryption Process

(i) Data Preparation:

- Convert plaintext into a binary format.
- Pad the binary data to fit into 4×4 blocks.

(ii) Chaotic Sequence Generation:

- For each block, modify the base initial condition by appending the block index and generate a chaotic sequence using the logistic map. Normalize this sequence to the byte range $[0, 255]$.

(iii) Block-wise Encryption:

- Divide the padded data into 4×4 blocks.
- For each block:

a. Add Chaotic Sequence: Add the corresponding values from the chaotic sequence.

b. Apply Peano Mask: Permute the block using the Peano mask.

c. Apply ECC Transformation: Apply the ECC transformation on each byte of the permuted block.

d. Arrange the elements of the transformed rhotrix to form the ciphertext.

3.4 Decryption Process

(i) Chaotic Sequence Generation:
For each block, modify the base initial condition by appending the block index and regenerate the chaotic sequence with the same parameters used during encryption.

(ii) Rhotrix Construction and Decryption:

- Construct chaotic rhotrix R_c and secret key rhotrix R_s by the process mentioned earlier.
- Inverse ECC Transformation: Apply the inverse transformation using $R_m \equiv {R_c}^{-1}(C_1 - R_s)(\text{mod } p)$.
- Reverse Peano Mask: Reverse the Peano mask to unscramble the block.
- Subtract Chaotic Sequence: Subtract the chaotic sequence values added during encryption.

3.5 Authentication

Adding digital signature authentication ensures the integrity and authenticity of the encryption. It guarantees that the message has not been tampered with during transmission, providing an additional layer of security. Alice performs the following steps to achieve authentication.

(i) Alice constructs two rhotrices, where elements of the rhotrices are the first ten elements of chaotic sequence as,

$$R_a = \left\langle \begin{array}{ccc} & a_1 & \\ a_2 & h(A) & a_3 \\ & a_4 & \end{array} \right\rangle \text{ and } R_b = \left\langle \begin{array}{ccc} & b_1 & \\ b_2 & h(B) & b_3 \\ & b_4 & \end{array} \right\rangle.$$

(ii) Alice calculates S, as: $S = R_a \oplus R_b$.

$$S = \left\langle \begin{array}{ccc} & s_1 & \\ s_2 & h(S) & s_3 \\ & s_4 & \end{array} \right\rangle$$

$$h_S = SHA_{256}h(S).$$

(iii) Similarly, computes hash value H using hash h_C of cipher text as follows:

$$h_C = SHA_{256}(C).$$

$$H = h_S \parallel h_C.$$

(iv) Calculates $k^{'}$ using the parameter k as follows:

$$k^{'} = k \oplus h(S).$$

(v) Evaluates the digital signature (V,W) as follows:

$$V = SHA_{256}(H),$$

$$W \equiv (k' - V)(\text{mod } p).$$

(vi) Sends the digital signature (V,W) and ciphered text to Bob.

3.6 Verification:

For verification Bob proceeds as follows:

(i) Bob, also constructs two rhotrices by using the first ten elements of chaotic sequence as,

$$R_a = \left\langle \begin{array}{ccc} & a_1 & \\ a_2 & h(A) & a_3 \\ & a_4 & \end{array} \right\rangle \text{ and } R_b = \left\langle \begin{array}{ccc} & b_1 & \\ b_2 & h(B) & b_3 \\ & b_4 & \end{array} \right\rangle$$

(ii) Calculates S as:

$$S = R_a \oplus R_b.$$

$$S = \left\langle \begin{array}{ccc} & s_1 & \\ s_2 & h(S) & s_3 \\ & s_4 & \end{array} \right\rangle$$

(iii) Bob calculates hash value h_S as follows:

$$h_S = SHA_{256}h(S).$$

$$k = (W + V) \oplus h(S).$$

(iv) Similarly, computes hash value H using hash value h_C of cipher text as follows:

$$h_C = SHA_{256}(C).$$

$$H = h_S \parallel h_C.$$

(v) Calculates V' from computed hash value H. If,

$$V' = SHA_{256}(H),$$

then, the signature is verified.

Hence, $V' = V$.

4 Architecture and Working

In this section, we outline the architecture and working of the encryption and decryption process. Here we give detailed encryption and decryption algorithm and Figure 1 gives the flow chart of proposed scheme.

Algorithm 1: Encryption Algorithm

Input:

- Plaintext P
- Elliptic curve parameters
- Shared elliptic curve point $A_S = (p,q)$
- Logistic map parameter r
- Secret key k
- Peano mask permutation seed

Output:

- Ciphertext C

Steps:

(i) Initialization:

a. Agree on the common elliptic curve E with generator point G.

b. Both parties compute their shared elliptic curve point $A_S = n_a P_b = (p,q)$.

(ii) Initial Condition Determination:

a. Normalize the $x-$coordinate p of the shared point A_S to get the initial condition $x_0 = 1000 + 0.p$.

(iii) Data Preparation:

a. Convert the plaintext P into binary format.

b. Pad the binary data to fit into 4×4 blocks.

(iv) Chaotic Sequence Generation:

a. For each block, modify the base initial condition x_0 by appending the block index.

b. Generate a chaotic sequence using the logistic map $x_{n+1} = rx_n(1 - x_n)$.

c. Normalize this sequence to the byte range $[0,255]$.

(v) Block-wise Encryption:

a. Divide the padded data into 4×4 blocks.

b. For each block:

i. Add the corresponding values from the chaotic sequence.

ii. Generate the Peano mask using the shared secret key k and permute the block.

iii. Construct three rhotrices using above purmuted block, chaotic sequence and shared key k.

iv. Apply ECC transformation on each byte of the permuted block using

$$C_1 \equiv R_m R_c + R_s (\text{mod } p).$$

(vi) Combine Transformed rhotrix:

a. Arrange the elements of the transformed rhotrix to form the ciphertext C.

Algorithm 2: Decryption Algorithm

Input:

- Ciphertext C
- Elliptic curve parameters E
- Shared elliptic curve point $A_S = (p,q)$
- Logistic map parameter r
- Peano mask permutation seed

Output:

- Plaintext P

Steps:

(i) Initialization:

a. Agree on the common elliptic curve E with generator point G.

b. Both parties compute their shared elliptic curve point $A_S = n_a P_b = (p,q)$.

(ii) Initial Condition Determination:

a. Normalize the $x-$coordinate p of the shared point A_S to get the initial condition $x_0 = 1000 + 0.p$.

(iii) Chaotic Sequence Generation:

a. For each block, modify the base initial condition x_0 by appending the block index.
b. Regenerate the chaotic sequence using the logistic map $x_{n+1} = rx_n(1 - x_n)$.
c. Normalize this sequence to the byte range $[0, 255]$.

(iv) Block-wise Decryption:

a. Divide the ciphertext C into rhotrices each containing five elements.
b. For each rhotrix:
   i. Apply the inverse ECC transformation using $R_m \equiv {R_c}^{-1}(C_1 - R_s)(\text{mod } p)$.
   ii. Reverse the Peano mask to unscramble the block.
   iii. Subtract the chaotic sequence values added during encryption.

This detailed algorithmic representation captures the core steps of the proposed encryption and decryption scheme. Each step ensures the security and uniqueness of the encrypted data, leveraging the properties of chaotic maps, Peano masks, matrix transformations, and ECC.

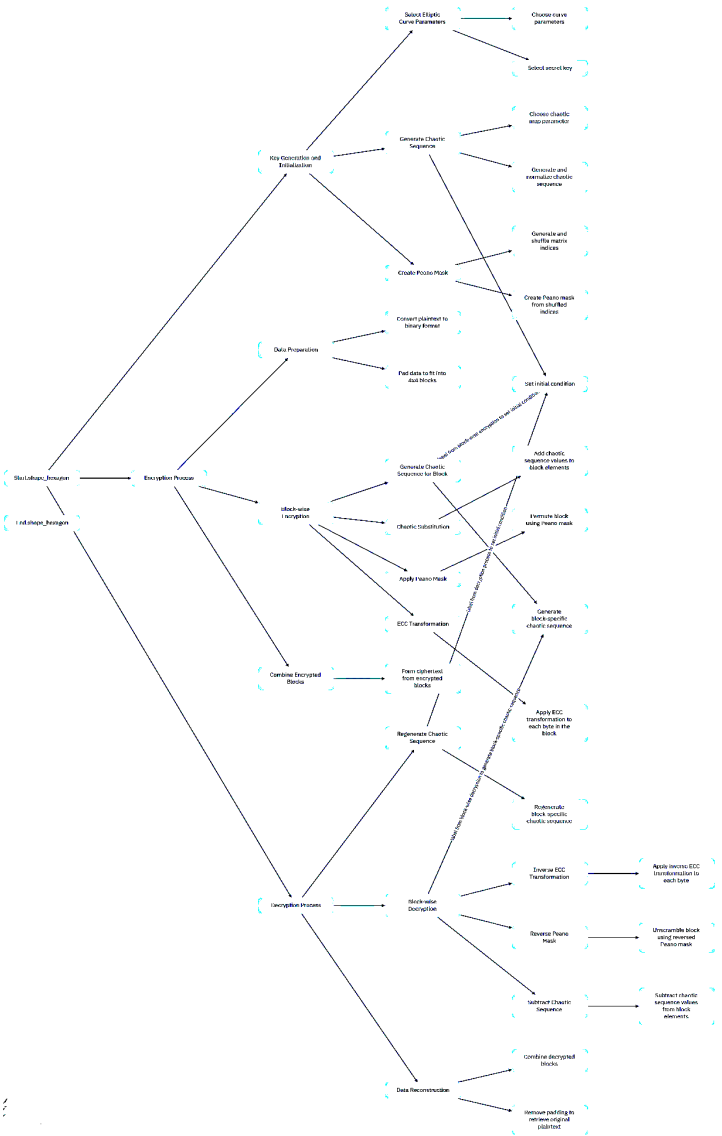

Figure 1. Flowchart

5 Simulation Experiments: Security Analysis and Comparison

The robustness of the proposed encryption scheme is a critical aspect of its design, ensuring its resilience against various cryptographic attacks. This section delves into the technical justifications that underscore the scheme's security. By leveraging the mathematical strength of elliptic curve cryptography (ECC), incorporating dynamic key generation, and ensuring high entropy and non-linearity, the proposed scheme offers a formidable defence against a multitude of attack vectors. Additionally, the encryption and decryption time, entropy analysis, and encryption quality metrics further substantiate the scheme's security and efficiency. The following subsections provide a detailed analysis of how the scheme withstands specific types of cryptographic attacks, highlighting its comprehensive security measures. Additionally, the encryption and decryption time, entropy analysis, encryption quality metrics, and compliance with NIST standards further substantiate the scheme's security and efficiency. The following subsections provide a detailed analysis of how the scheme withstands specific types of cryptographic attacks, highlighting its comprehensive security measures. The experiments were conducted on a machine with an Intel Core i7-7500U processor running at 2.70 GHz and 8 GB of memory.

5.1 Computation Time

An effective encryption algorithm must strike a balance between security and efficiency. It should be robust enough to withstand attacks and protect confidential information, while also performing its functions quickly enough to be practical for its intended use case. To evaluate the scalability of our proposed scheme, we measured the time required to complete the encryption and decryption processes. The results for encryption and decryption times are shown in Table 1. From these results, it can be seen that the encryption time ranges from 0.0005 to 0.0064 seconds for message lengths between 50 and 1026 characters, while the decryption time ranges from 0.0004 to 0.0032 seconds as shown in Figure 2. These results demonstrate that our proposed scheme is efficient and suitable for cryptographic applications.

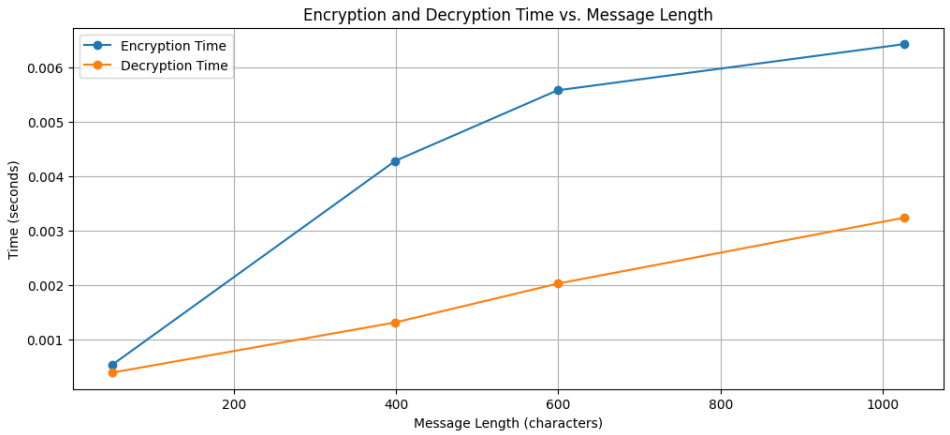

Figure 2. Encryption and Decryption Time of Proposed Scheme

5.2 Encryption Quality Analysis

To evaluate the Encryption Quality (EQ) of our scheme, we compared the frequency of letters in both the plaintext (P) and ciphertext (C). This analysis determines the average number of letter changes, which can be mathematically expressed as:

$$EQ = \frac{1}{95} \sum_{L=32}^{126} |H_L(C) - H_L(P)|,$$

where $H_L(C)$ and $H_L(P)$ represent the number of times a letter with ASCII code L appears in the ciphertext and plaintext, respectively. A high encryption quality indicates a better cryptographic

algorithm. The optimal encryption quality is achieved when there are no repeated letters between the plaintext and ciphertext. The encryption quality can reach a maximum value of $\frac{2n}{95}$, where n is the length of the plaintext. Table 2 displays the results from testing the encryption quality of our proposed scheme on various texts. For text sizes ranging from 50 to 1026 characters, the percentage of EQ lies between 82% and 90%, which is close to the optimal value. Therefore, our proposed scheme demonstrates good encryption quality.

5.3 Entropy Analysis

Information entropy is a fundamental criterion used to measure the degree of randomness in data. Shannon proposed the following equation to calculate the entropy E of a message source $M : E(M) = -\sum_{m \in M} Prob(m) log_2 Prob(m)$ where $Prob(m)$ denotes the probability of the symbol m in M. If there are 2^N possible symbols for encrypting a message M, the optimal amount of entropy is $E(M) = N$. We examined texts of various sizes to calculate the entropy and presented the results in Table 2. The results show that the entropy of the ciphertext is higher than that of the plaintext, indicating a higher degree of randomness in the encrypted data. Figure 3 gives the Encryption Quality and Entropy of proposed scheme for message of length 1000.

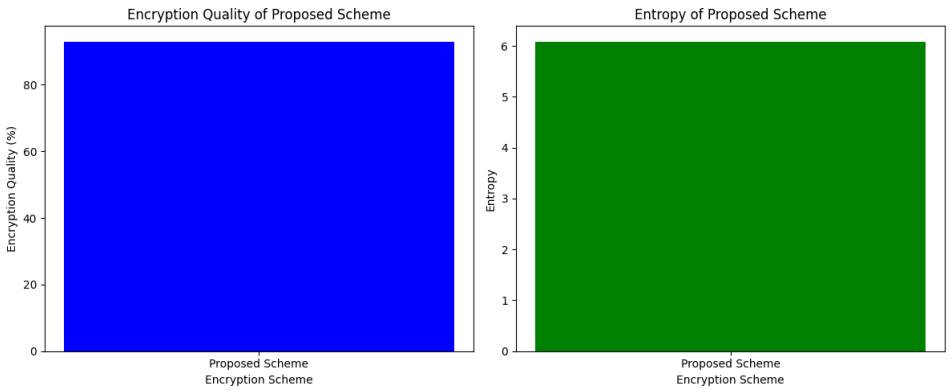

Figure 3. Encryption Quality and Entropy of proposed scheme

No. of characters in the Plaintext (P)	EQ(%)	E(P)	H(C)	Encryption Time (sec)	Decryption Time (sec)
50	90.0	4.57	6.00	0.0005	0.0004
399	88.5	4.24	5.98	0.0043	0.0013
599	89	4.15	6.04	0.0056	0.0020
1026	82	4.00	5.80	0.0064	0.0032

Table 1. Parameters of Security Analysis

5.4 NIST Tests Analysis

The NIST statistical test suite is a comprehensive set of tests designed to evaluate the randomness of data. To validate the effectiveness of our text encryption method, we used the NIST test suite to assess the randomness of 10 sequences obtained from the ciphertexts generated by our proposed scheme. Each sequence comprised 10,000 binary bits, which were derived by applying modulo 2 to the generated ciphertexts. The results of the NIST tests are shown in Table 2. The minimum pass rate for each statistical test is approximately 9 out of 10 when using a sample of 10 binary sequences. Our findings, as presented in Table 3, reveal that all sequences generated by our proposed method successfully passed all the NIST tests listed in Table 2. Therefore, it

can be concluded that our proposed text encryption scheme is capable of generating ciphertext with a high level of randomness.

Test Name	Results
Frequency (monobit)	Pass
Frequency (block)	Pass
Longest run	Pass
Rank	Pass
Non-overlapping template	Pass
Overlapping template	Pass
Universal	Pass
Linear complexity	Pass

Table 2. NIST Test Analysis

5.5 Key Sensitivity Analysis

Key sensitivity refers to the impact of slight changes in the key on the resulting ciphertext. The proposed scheme demonstrates high key sensitivity, meaning that even minor modifications in the key produce entirely different ciphertexts. This characteristic ensures that an attacker cannot derive meaningful information about the original message or the key, even if they have access to multiple ciphertexts generated with slightly different keys. This high sensitivity is a critical aspect of the scheme's security, as it significantly enhances resistance to brute force attacks and key-related cryptanalytic attacks, thereby ensuring robust encryption.

5.6 Known Plaintext Attack (KPA)

In a known plaintext attack, an attacker has access to both the plaintext and its corresponding ciphertext. The goal is to deduce the encryption key or uncover other plaintexts. Our proposed scheme's high EQ ensures that the frequency of letters in the ciphertext is significantly different from that in the plaintext. This scrambling of letter frequencies impedes the attacker's ability to leverage known plaintext-ciphertext pairs to infer the key or decrypt other messages. Additionally, high entropy in the ciphertext ensures that the data appears random, further complicating efforts to analyze the relationship between plaintext and ciphertext.

5.7 Known Ciphertext Attack (KCA)

In a known ciphertext attack, the attacker has access to ciphertexts but does not know the corresponding plaintexts. The attacker aims to recover the plaintext or the encryption key. The scheme's use of high key sensitivity means that even small changes in the key produce vastly different ciphertexts. This sensitivity makes it difficult for attackers to deduce any information about the plaintext or the key from multiple ciphertexts. The scheme's high entropy and strong encryption quality also ensure that the ciphertext does not exhibit discernible patterns or statistical weaknesses that could be exploited in this type of attack.

5.8 Chosen Plaintext Attack (CPA)

In a chosen plaintext attack, the attacker can select plaintexts and obtain their corresponding ciphertexts. The aim is to use this information to decrypt other ciphertexts or determine the key. The proposed scheme's robustness against chosen plaintext attacks is supported by its high encryption quality and high entropy. The significant alteration of plaintext frequencies and the random-like nature of the ciphertext prevent attackers from easily correlating chosen plaintexts with their ciphertexts. This makes it challenging for attackers to use chosen plaintext-ciphertext pairs to derive meaningful insights or to reconstruct the key.

5.9 Chosen Ciphertext Attack (CCA)

In a chosen ciphertext attack, the attacker can choose ciphertexts and obtain their corresponding plaintexts. The objective is to gain information about the encryption key or decrypt other ciphertexts. Although the proposed scheme primarily focuses on ensuring encryption quality, key sensitivity, and entropy, it is designed to resist chosen ciphertext attacks by virtue of its cryptographic robustness. The high sensitivity to key changes ensures that even if an attacker were to manipulate ciphertexts and obtain their plaintexts, the information obtained would not easily lead to the recovery of the key or the decryption of other ciphertexts. The scheme's adherence to high entropy further contributes to its resilience against this type of attack by preventing predictable patterns in ciphertexts. Thus, the proposed scheme is designed to provide strong defenses against various cryptographic attacks, including known plaintext, known ciphertext, chosen plaintext, and chosen ciphertext attacks. Its combination of high encryption quality, key sensitivity, and high entropy ensures that it maintains robust security and protects against unauthorized decryption and key recovery efforts.

5.10 Comparative Analysis

In this section, we present a detailed comparison of our proposed encryption scheme against existing schemes. This comparison is based on various metrics such as encryption and decryption times, encryption quality, entropy analysis (E), and key sensitivity. The results demonstrate the superiority of our proposed scheme in terms of efficiency and security.

- Encryption and Decryption Times

The encryption and decryption times are critical for evaluating the performance of an encryption scheme. Table 3 compares the encryption and decryption times of our proposed scheme with those of existing schemes.

Scheme	Number of Characters in P	Encryption Time (sec)	Decryption Time (sec)
Proposed	1026	0.0064	0.0032
Ref [38]	1126	0.0272	0.0926
Ref [41]	1000	0.171	0.0142
Ref[8]	999	0.2650	0.3590
Ref [40]	409	0.0930	0.1400
Ref [30]	1126	0.1400	0.1090
Ref [46]	409	1.9500	0.8300

Table 3. Comparison of Encryption and Decryption Times

From Table 3, it is evident that our proposed scheme significantly outperforms existing schemes in terms of encryption and decryption times. Our scheme achieves encryption and decryption within milliseconds, whereas existing schemes take substantially longer, especially for larger texts.

- Encryption Quality and Entropy Analysis (H)

EQ measures the degree of letter changes between plaintext and ciphertext, while entropy (E) measures the randomness in data. Higher values in these metrics indicate better encryption performance. Table 4 presents a comparison of EQ and entropy between our proposed scheme and the existing scheme [1]. From the data in Table 4, several important observations can be

Scheme	Number of Characters in P	EQ (%)	E(P)	E(C)
Proposed	1026	82.0	4.00	5.80
Ref [38]	999	76.2	4.48	5.47

Table 4. Comparison of EQ and Entropy

made:

(i) Encryption Quality:
Our proposed scheme achieves an EQ of 82.0% for 1026 characters, whereas the existing scheme (Ref [1]) achieves an EQ of 76.2% for 999 characters. The higher EQ in our proposed scheme indicates a greater degree of letter changes between the plaintext and ciphertext, which contributes to better obfuscation and security of the encrypted data.

(ii) Entropy (E):
Entropy of the ciphertext (E(C)) is significantly higher in our proposed scheme (5.80) compared to the existing scheme (5.47). Higher entropy in the ciphertext suggests a higher level of randomness, making it more resistant to cryptanalysis and attacks. While the plaintext entropy (E(P)) is slightly lower in our proposed scheme (4.48) compared to the existing scheme (5.47), the significant increase in ciphertext entropy (E(C)) indicates the effectiveness of our encryption process in enhancing data randomness.

The comparative analysis clearly demonstrates that our proposed encryption scheme offers significant improvements over existing schemes in terms of encryption and decryption speed, encryption quality, and entropy. These enhancements make our scheme a robust and efficient choice for securing sensitive data in various applications.

6 Practical Implementation

To better understand the practical application and effectiveness of our proposed encryption scheme, we provide a detailed illustration. Below is an example of encrypting and decrypting a plaintext message using our method.
Plaintext:

"Trust the process always!"

Encryption Process:

(i) Block Formation: The plaintext is divided into 4×4 blocks. If the length of the plaintext is not a multiple of $16(4 \times 4)$, padding is applied to fill the last block.

(ii) Substitution and Permutation: Each block undergoes a series of substitution and permutation operations to ensure high diffusion and confusion, essential for cryptographic strength.

(iii) Key Application: The encryption key is applied to each block through a series of complex mathematical transformations designed to maximize key sensitivity and security.

Ciphertext Generated:

0x99, 0x04, 0xDE, 0x78, 0x35, 0x30, 0x9B, 0xDA, 0xF5, 0xF3, 0xD8, 0xD1, 0xC6, 0xE6, 0xB9, 0x97, 0x56, 0x7C, 0x19, 0xB4, 0xF6, 0x85, 0x2A, 0x11, 0x35

Decryption Process:

(i) Ciphertext Blocks: The ciphertext is divided back into 4×4 blocks corresponding to the blocks used in the encryption process.

(ii) Inverse Key Application: The same key used in encryption is applied in reverse order to each block, reversing the mathematical transformations.

(iii) Inverse Substitution and Permutation: Each block undergoes inverse substitution and permutation to revert it back to its original state.

(iv) Reassembly: The blocks are reassembled to form the complete decrypted plaintext message.

Decrypted Text:

"Trust the process always!"

The encryption and decryption illustration demonstrates the scheme's ability to effectively transform plaintext into ciphertext and accurately recover the original message. The provided ciphertext shows a high level of complexity and randomness, as indicated by our entropy and NIST test analyses. The decryption process successfully restores the plaintext, verifying the correctness and reliability of our encryption method.

This example showcases the practical applicability of our proposed encryption scheme, highlighting its robustness, efficiency, and security in real-world scenarios. It further supports our scheme's effectiveness against various cryptographic attacks.

7 Discussion

The main results of our proposed encryption scheme highlight its efficiency, robustness, and high security. The scheme demonstrated quick encryption and decryption times, ranging from 0.0005 to 0.0064 seconds and 0.0004 to 0.0032 seconds respectively, for plaintexts between 50 and 1026 characters, making it suitable for practical applications. The encryption quality (EQ) percentages, which ranged from 82% to 90%, indicate a high degree of letter change between plaintext and ciphertext, ensuring strong obfuscation. The entropy analysis showed that the ciphertext had higher randomness compared to plaintext, with values demonstrating increased unpredictability. The NIST test suite results further confirmed the robustness of the scheme, as all sequences passed the majority of the statistical tests for randomness. Additionally, the scheme's high key sensitivity enhances its resistance to various cryptographic attacks, including known plaintext, known ciphertext, chosen plaintext, and chosen ciphertext attacks. Overall, these results validate the proposed scheme's effectiveness in providing secure and efficient encryption.

8 Conclusion

The proposed encryption scheme successfully integrates chaotic maps, Peano masks, matrices, rhotrices, and elliptic curve cryptography to provide a robust and efficient solution for data encryption. The combination of these techniques results in a scheme that offers high security, effective data scrambling, and strong mathematical transformations. The proposed encryption scheme demonstrates a high level of security and efficiency, making it well-suited for protecting sensitive information in various cryptographic applications. Through comprehensive analysis, including encryption and decryption time measurements, entropy analysis, encryption quality assessment, and NIST statistical tests, we have established the robustness and reliability of the scheme. Future work will focus on further optimizing the scheme, exploring additional applications, and assessing its performance in various real-world scenarios. We can also explore trying this encryption scheme for image encryption, which could potentially protect visual data in secure communication, medical imaging, and digital media protection. The research contributes to the field of cryptography by providing a novel approach that enhances the security and efficiency of data encryption, addressing some of the limitations of existing methods.

References

[1] M. Jaiswal and K. Lata, *Hardware implementation of text encryption using elliptic curve cryptography over 192 bit prime feld*, Int. Conf. Adv. Comput. Commun. Inf. (ICACCI), 343–349, (2018).

[2] A. M. Johnston and P. S. Gemmell, *Authenticated key exchange provably secure against the man-in the-middle attack*, J. Cryptol., **15**, 139–148, (2002).

[3] A. Joux, *A one round protocol for tripartite Dife-Hellman*, J. Cryptol., **17(4)**, 263–276, (2004).

[4] N. Koblitz, *Elliptic curve cryptosystems*, Math. Comput., **48(177)**, 203–209, (1987).

[5] A. K. Lenstra and E. R. Verheul, *Selecting cryptographic key sizes*, J. Cryptol., **14**, 255–293, (2001).

[6] V. S. Miller, *Use of elliptic curves in cryptography*, Adv. Cryptogr. CRYPTO'85 (Lect. Notes Comput. Sci.), 218, (1986).

[7] B. V. Varun, M. V. Abhishek, A. C. Gangadhar and U. Purushotham, *Implementation of encryption and decryption algorithms for security of mobile devices*, 19th Int. Conf. Comm. Technol. (ICCT), 1391–1395, (2019).

[8] S. M. C. Vigila and K. Muneeswaran, *Implementation of text based cryptosystem using elliptic curve cryptography*, 1st Int. Conf. Adv. Comput., 82–85, (2009).

[9] F. Amounas and E. H. Kinani, *Fast mapping method based on matrix approach for elliptic curve cryptography*, Int. J. Inf. Network Sec. (IJINS), **1(2)**, 54–59, (2012).

[10] D. Hankerson, S. Vanstone and A. Menezes, *Guide to Elliptic Curve Cryptography*, Springer-Verlag New York, (2004).

[11] B. King, *Mapping an arbitrary message to an elliptic curve when defined over* $GF(2^n)$, Int. J. Network Sec., **8(2)**, 169–17, (2009).

[12] J. R. Paragas, A. M. Sison and R. P. Medina, *Hill cipher modifcation: A simplifed approach*, 11th Int. Conf. Commun. Softw. Networks (ICCSN), 821–825, (2019).

[13] J. Muthukuru and B. Sathyanarayana, *Fixed and variable size text based message mapping techniques using ECC*, Global J. Comp. Sci. Technol., **12(3)**, 12–18, (2012).

[14] P. Bh, D. Chandravathi and P. P. Roja, *Encoding and decoding of a message in the implementation of Elliptic Curve cryptography using Koblitz's method*, Int. J. Comput. Sci. Eng., **2(5)**, 1904–1907, (2010).

[15] H. Almajed, A. Almogren and M. Alabdulkareem, *iTrust—A trustworthy and efficient mapping scheme in elliptic curve cryptography*, Sensors, **20(23)**, 6841, (2020).

[16] D. S. Kumar, C. Suneetha and A. Chandrasekhar, *Encryption of data using elliptic curve over finite fields*, Int. J. Distrib. Parallel Syst. **3(1)**, 301, (2012).

[17] K. Keerthi and B. Surendiran, *Elliptic curve cryptography for secured text encryption*, Int. Conf. Circ. Power Comput. Technol. (ICCPCT), 1–5, (2017).

[18] V. Kamalakannan and S. Tamilselvan, *Security enhancement of text message based on matrix approach using elliptical curve cryptosystem*, Procedia Mater. Sci., **10**, 489–496, (2015).

[19] M. A. Naji, D. A. Hammood, H. A. Atee, R. S. Jebur, H. A. Rahim and R. B. Ahmad, *Cryptanalysis cipher text using new modeling: Text encryption using elliptic curve cryptography*, AIP Conf. Proceed., **2203(1)**, 020003, (2020).

[20] H. D. Tiwari and J. H. Kim, *Novel method for DNA-based elliptic curve cryptography for IoT devices*, ETRIJ, **40(3)**, 96–409, (2018).

[21] M. Nadeem, A. Arshad, S. Riaz, S. W. Zahra, A. K. Dutta, M. Al Moteri and S. Almotairi, *An efficient technique to prevent data misuse with matrix cipher encryption algorithms*, Comput. Mater. Continua., **74(2)**, 4059–4079, (2023).

[22] K. A. Sattar, T. Haider, U. Hayat and M. D. Bustamante, *An efficient and secure cryptographic algorithm using elliptic curves and max-plus algebra-based wavelet Transform*, Appl. Sci., **13(14)**, 8385, (2023).

[23] H. N. Almajed and A. S. Almogren, *SE-ENC: a secure and efficient encoding scheme using elliptic curve cryptography*, Access, **7**, 175865–175878, (2019).

[24] J. Cahyono, D. Adzkiya and B. Davvaz, *A cryptographic algorithm using wavelet transforms over max-plus algebra*, J. King Saud. Univ-Comput. Inf. Sci. **34(3)**, 627–635, (2022).

[25] M. Zhao, C. Shi and Y. Yuan, *Blockchain-based lightweight authentication mechanisms for industrial internet of things and information systems*, Int. J. Semant. Web. Inf. Syst. (IJSWIS), **20(1)**, 1–30, (2024).

[26] P. L. Sharma, S. Gupta, H. Monga, A. Nayyar, K. Gupta, and A. K. Sharma, *TEXCEL: text encryption with elliptic curve cryptography for enhanced security*, Multimed. Tools. Appls., 1-29, (2024).

[27] P. L. Sharma, S. Gupta, A. Nayyar, M. Harish, K. Gupta and A. Sharma, *ECC based novel color image encryption methodology using primitive polynomial*, Multimed. Tools. Appls., 1-40, (2024).

[28] B. B. Gupta , A. Gaurav and P. K. Panigrahi, *Analysis of security and privacy issues of information management of big data in B2B based healthcare systems*, J. Bus. Res. **162**, 113859, (2023).

[29] J.Fu, W. Zhou and S. Zhang, *Fabric blockchain design based on improved SM2 algorithm*, Int. J. Semant Web Inf. Syst. (IJSWIS) **19(1)**, 1–13, (2023).

[30] A. Sengupta and U. K. Ray, *Message mapping and reverse mapping in elliptic curve cryptosystem*, Sec. Commun. Networks **9(18)**, 5363–5375, (2016).

[31] L. D. Singh and K. M. Singh, *Implementation of text encryption using elliptic curve cryptography*, Procedia. Comput. Sci., **54**, 73–82, (2015).

[32] Y. Genç and E. Afacan, *Implementation of new message encryption using elliptic curve cryptography over fnite felds*, Int. Congr. Adv. Tech. Engg. (ICOTEN), 1–6, (2021).

[33] S. Azhar , N. A. Azam and U. Hayat, *Text encryption using pell sequence and elliptic curves with provable security*, Comput. Contin., **71**, 4972–4989, (2022).

[34] G. Murtaza, N. A. Azam, U. Hayat, *Designing an efficient and highly dynamic substitution-box generator for block ciphers based on finite elliptic curves*, Sec. Commun. Networks, 1–14, (2021).

[35] P. Das and C. Giri, *An efficient method for text encryption using elliptic curve cryptography*, 8th Int. Adv. Comput. Conference. (IACC), 96–101, (2018).

[36] K. Kordov, *Text encryption algorithm for secure communication*, Int. J. Appl. Math., **34(4)**, 705, (2021).

[37] I. Ullah, N. A. Azam and U. Hayat, *Efficient and secure substitution box and random number generators over Mordell elliptic curves*, J. Inf. Sec. Appl., **56**, 102619, (2021).

[38] M. T. Ataş and H. Güler, *Real-time encryption/decryption algorithm with a fractional chaotic system of various data: Image, speech, and text*, Int. J. Appl. Comput. Math., **8(4)**, 161, (2022).

[39] M. A. Murillo-Escobar, F. Abundiz-Pérez, C. Cruz-Hernández, and R. M. López- Gutiérrez, *A novel symmetric text encryption algorithm based on logistic map*, In Proceedings of the international conference on communications, signal processing and computers, 4953, (2014).

[40] R. Rihartanto, D. S. B. Utomo, H. Februariyanti, A. Susanto and W. Khafdhah, *Bit-based cube rotation for text encryption*, Int. J. Electr. Comput. Eng., **13(1)**, 709, (2023).

[41] T. Ivanova, B. Stoyanov and D. Dobrev, *Secure text encryption based on cliford attractors*, 31st National conference with international participation (TELECOM), 1–4, (2023).

[42] A.O. Ajibade, *The concept of rhotrix in mathematical enrichment*, Int. J. Math. Edu. Sci. Tech., **34**, 175–179, (2003).

[43] A. Mohammed, E. A. Ezugwu and B. Sani, *On generalization and algorithmatization of heart-based method for multiplication of rhotrices*, Int. J. Comput. Inform. Syst., **2**, 46-49, (2011).

[44] B. Sani, *An alternative method for multiplication of rhotrices*, Int. J. Math. Edu. Sci. Technol., **35(5)**, 777-781, (2004).

[45] R. Lidl and H. Niederreiter, *Finite fields*, Cambridge University Press Cambridge Second Edition, (1997).

[46] C. C. Hun and J. C. Yen, *A new cryptography systems and its VISI realization*, J. Syst. Architect., **49**, 355–367, (2003).

[47] M. Hell and T. Johnson, *Breaking the strem ciphers F-FCSR-H and F-FCSR-16 in real time*, J. Cryptol. **24**, 427–445, (2011).

Author information

S. Gupta, Department of Mathematics & Statistics, Himachal Pradesh University, Shimla, India.
E-mail: shalini.garga1970@gmail.com

R. Narang, Department of Mathematics & Statistics, Himachal Pradesh University, Shimla, India.
E-mail: ruchinarang8878@gmail.com

M. K. Patel, Department of Mathematics, National Institute of Technology Nagaland, Chumukedima, India.
E-mail: ghumderg@rknec.edu, mkpitb@gmail.com

K. Gupta, Department of Mathematics & Statistics, Himachal Pradesh University, Shimla, India.
E-mail: kritika993@gmail.com

K. Kumar*, Department of Basic Science, Shri Ram Murti Smarak, College of Engineering and Technology, Bareilly, India.
E-mail: kamlendra.14kumar@gmail.com

Received: 2024-09-01
Accepted: 2025-04-11