Elliptic Key and Curve Diffie-Hellman with Magic Square for Secure Image Encryption

B. Vybhavi [1,*], K. Mani [1], V. Uma Devi [1], and S. Kumar Chandar [2]

[1] Department of Computer Science, Nehru Memorial College (Autonomous), Affiliated to Bharathidasan University, Puthanampatti, Tiruchirappalli, India

[2] School of Business and Management, Christ University, Bangalore, Karnataka, India

Email: vybhavib2@gmail.com (B.V.); nitishmanik@gmail.com (K.M.); yazh1999@gmail.com (V.U.D.); kumar.chandar@christuniversity.in (S.K.C.)

*Corresponding author

***Abstract*—Image encryption technique is the procedure of converting an original image into an unreadable form to ensure privacy, security, and confidentiality during transmission. The Magic Square (MS) is a structured matrix in which all horizontal, vertical, and diagonal operations are applied to generate a deterministic, difficult pixel rearrangement pattern. However, the magic square and Elliptic Curve Cryptography (ECC)-based method do not perform well on color and high-resolution images, failing to achieve better integration between pixel-level and ECC-based key management, thereby minimizing the model's encryption efficiency. To overcome these limitations, this research proposes the Elliptic Key-elliptic Curve Diffie-Hellman (EK-CDH) with MS for generating secure and effective key generation and key exchange mechanisms. The Elliptic Curve Diffie-Hellman (ECDH) provides strong, reliable shared keys for color and high-resolution images by connecting ECDH with a magic square, ensuring better coordination between the pixel-level and key management. The EK is generated by ECDH, which enables the sender and receiver to securely evaluate a shared secret key over an untrusted channel. The proposed EK-ECDH with MS achieves low encryption and decryption times of 0.24 s and 0.17 s, respectively, when compared with the enhanced Royat model with ECC.**

***Keywords*—elliptic curve cryptography, elliptic curve Diffie-Hellman, image encryption, key exchange, magic square, pixel level**

I. Introduction

Images are the essential information that consists of large digital data with huge volume and redundancy, which challenges the model to store and transmit the image securely as compared to text data [1]. Image security is protected against different threats such as modification, eavesdropping, and illegal duplication. Image encryption schemes are utilized as an original form of the conventional encryption model, which converts the original image into a meaningless image [2]. The image encryption algorithm is partitioned into four classes. Image encryption relies on pixel replacement and matrix transformation [3]. The Elliptic Curve Cryptography (ECC) is a cryptographic approach that is used for mapping and encryption schemes, which generate a powerful framework for data integrity and encryption [4]. The image encryption approach confirms the integrity and confidentiality of image data [5]. Through operating an encryption model, the images are securely transferred, stored, and also utilized in various applications such as copyright protection, medical image transmission, secure image sharing, and cloud storage [6]. Additionally, the image information continues to increase in size, and challenges effective encryption and decryption performance [7]. The encryption algorithms were established for estimating various types of images such as color, grayscale, and medical images, which lie within space [8].

The Magic Square (MS) transformation is a matrix transform encryption model that relies on encryption by altering the pixel-level position [9]. The original histogram image does not change after the magic square transformation. While MS undergoes displacement, the encrypted images are exploited by breaches [10]. The MS image encryption algorithm relies on key sharing and attains a high level of security, whereas key-sharing-based encryption would not be appropriate for images with a large amount of data [11]. Conventional encryption approaches, such as symmetric and asymmetric encryption, are not directly used in the image because of unique characteristics, namely, complex structures, huge file size, and high redundancy [12]. The image encryption that depends on chaos is classified into chaotic image encryption, which relies on pixel gray-level transformation, thereby scrambling and hiding [13]. Scramblings for scrambling-based chaotic image encryption involve a scheme that relies on the Arnold transformation and MS transformation, which reduces the struggle of interference and affects the security [14]. Chaotic encryption depends on pixel gray-level transformation that alters original pixel values and makes it more efficient than simple position-based scrambling,

Manuscript received January 23, 2026; revised February 13, 2026; accepted March 27, 2026; published November 13, 2026.

which leads to low computational efficiency [15]. The chaotic image encryption by hiding operates an image integration approach to embed the cipher text into the template image by incorporating gray levels that efficiently encrypt the image with the other cover image [16]. Unlike the conventional hybrid encryption schemes, where the spatial permutation is fixed or pseudo-randomly generated. The proposed framework binds the magic square construction directly to the ECDH-derived session key. This mechanism ensures that each encryption instance produces a unique, key-dependent permutation structure. It significantly increases the resistance to known-plaintext and chosen-plaintext attacks, mainly for high-resolution and color images. The main contributions of this research are described below:

- The proposed Elliptic Key-Elliptic Curve Diffie-Hellman (EK-ECDH) with an MS model increases image encryption and effectively manages color and high-resolution images, generating strong security and minimizing pixel correlation.
- The EK is generated by ECDH, which assists in the secure shared key to handle magic square-based pixel scrambling, which ensures that only authorized users can encrypt or decrypt the image.
- The MS is utilized in image encryption to rearrange pixel positions to provide confusion and minimize the correlations between pixels. While integrated with EK, it enables effective and secure encryption for color and high-resolution images.

The research paper is organized as follows: Section II represents related works. Section III provides the proposed methodology. Section IV denotes experimental results and a discussion of this research. Section V concludes the paper.

II. Related Works

This section denotes an analysis of existing approaches using image encryption techniques, which highlights the advantages and limitations described below.

Kumar and Sharma [17] developed Arnold's cat map with ECC and Genetic Algorithm (GA) for increasing the quality of image encryption. An Arnold cat map was utilized to present chaos and randomness in the procedure of encryption, which consisted of shuffled pixel positions. The ECC encrypted pixel values through public and private keys, while GA was performed in a key generation procedure for increasing the encryption security scheme. The integration of the proposed model objective was to attain a high level of security and confidentiality with quality in image encryption. However, Arnold's cat map with ECC and GA struggled with computational overhead because of the association of ECC and GA, which affected the efficiency of devices.

Yadav and Singh [18] implemented an asymmetric authentication approach using three masks based on ECC with Fractional Hartley Domain (FHD), which was implemented to increase the image encryption security and avoid the vulnerabilities in cryptographic attacks. The combination of ECC with FHD to resist cryptographic attacks enhanced the security through the high resistance of ECC. The FHD was utilized to enhance the encryption key space and provide a large key space in the crypto system.

Pandey and Sharma [19] introduced the Logistic Piecewise Linear Chaotic Map (LPWLCM), and ECC was utilized to encrypt the image securely, and the encryption procedure was completed by GA. The ECC provided high security with small key sizes, while the encrypted image was resistant to unauthorized access. The GA was applied to optimize a ciphertext to attain maximum strength, image quality, and model performance. Nevertheless, the encryption procedure did not perform well for ultra-images, particularly in high-resolution video, which affected the model's efficiency.

Singh *et al*. [20] developed an enhanced Royat model with ECC, which was utilized for secure transmission between two communication parties, while a pixel grouping model was necessary for an entire encryption scheme. The Royat model was utilized for encrypting the input image, and the Secure Hash Algorithm (SHA256) was applied to generate a digital signature for the authentication procedure. The ECDH sharing approach was used for sharing the secret key while the sender and receiver operated on the shared key.

Rani *et al*. [21] implemented an encryption approach based on MS with variational encoding and a chaotic map for image encryption. The plain image coordination was randomized through the 2D Arnold scrambling algorithm in a confusion stage. The magic square was applied to modify the pixel values of the original image, while the encoding approach was performed to increase the high level of security. The piecewise linear chaotic map was employed in the diffusion stage to enhance randomization for increasing the complexity. However, the modification of image pixel values using the elements of the encryption approach MS decreased the encryption speed.

Sharma *et al*. [22] introduced an ECC-based color image encryption model using a primitive polynomial for encrypting the color and grayscale images. A primitive polynomial was used to map a plain image of pixel values, which showed output in a permutation. The shuffled pixels were diffused via a matrix attained by the Kronecker product of two invertible Suslin matrices. To increase security, an ECC approach associated a logistic chaotic map with an authentication of a digital signature. Nevertheless, the ECC encryption model required diffusion, which meant that the attacker modified one image, where the modifications affected the entire encrypted image and reduced the model's efficiency. While several hybrid image encryption schemes combine cryptographic key exchange with pixel permutation, which mainly relies on static or weakly randomized scrambling mechanisms. On the other hand, the proposed model uses elliptic curve-derived session keys directly in the permutation construction while creating a tightly coupled cryptographic-spatial framework. This enhances both unpredictability and scalability for high-resolution color images. Moreover, the security of the proposed model is evaluated under Kirchhoff's principle, where the

adversary is assumed to have full access to the encrypted image. In addition, the complete knowledge of the encryption algorithm, but no access to the private key is obtained with the help of Kirchhoff's principle. The attackers may attempt statistical, differential, and known-plaintext attacks, but do not have access to the private keys used in elliptic curve Diffie-Hellman key exchange. It is further assumed that a fresh session key is generated for each encryption instance. Under these assumptions, the proposed model works as a key-dependent permutation and encryption framework to prevent structural exploitation and key recovery.

III. Proposed Methodology

In the proposed methodology section, the proposed EK-ECDH with MS model increases image encryption, which effectively manages color and high-resolution images while generating strong security and minimizing pixel correlation. The proposed EK-ECDH with MS evaluates a highly difficult and secure block cipher system by associating ECC for key management with a magic square matrix and encryption at the pixel level. The elliptic curve is defined as $E_p(a, b)$ over a finite prime field $\mathbb{F}_p$. Let G denote the base point on the curve and n specify its order. The private keys of the communicating parties are specified using $d_A, d_B \in [1, n-1]$, while the corresponding public keys are defined as $Q_A = d_A G$ and $Q_B = d_B G$. The shared secret key is derived using the elliptic curve Diffie-Hellman that is denoted by K_S, respectively. The image $I \in \mathbb{R}^{H \times W \times C}$ is the original color of the image with height H, width W, and the color C channels. The encrypted image is denoted by I_e and the magic square-based permutation matrix is defined as P_{ms}. It is generated as a function of the shared key K_S. All the pixel indices are represented by using (i, j, c), where i, j and c correspond to spatial and channel indices. The ECC begins with the protocol, the sender, and the receiver, upon the elliptic curve parameters $\{a, b, p, G\}$. The DH protocol operates each party with a secure shared key point $K = (x, y)$ based on respective keys. The shared secret key is utilized to derive two components $K_1 = xG = (k_{11}, k_{12})$, and $K_2 = yG = (k_{21}, k_{222})$. The following subsequent arithmetic operations, consisting of key values and pixel data, are performed with a Galois field to ensure compatibility with the pixel range from 0 to 255. For encryption, plaintext images are decomposed into 10 blocks of message elements related to the magic square solution space. The 5×5 matrix is mentioned as S, 15 positions are initialized with key values from the shared secret K, and the remaining 10 positions hold the plaintext images, while the difficulties of selecting key positions are flexible but demand 15 various values. The ciphertext output is produced through the rules of the magic square, thereby calculating the columns, rows, and diagonal of S. The constraints 5×5 matrix produces 12 sums with two dependencies, the output system of $m = 10$, while a unique linear equation is generated. The diffusion of the encryption process is defined in Eq. (1), where each encrypted pixel value C_i is computed using modular arithmetic.

$$C_i = \sum_{S_{i,j}} \mathrm{mod}(256) \tag{1}$$

where C_i is the ciphertext sums of row, column, and key-dependent diffusion sequence are denoted as $S_{i,j}$, which is derived from the shared secret. The modulo operation with base 256 ensures that the resulting pixel intensity remains within the valid 8-bit grayscale or the color range. This operation ensures strong diffusion by making each encrypted pixel dependent on both the plaintext and the cryptographic key.

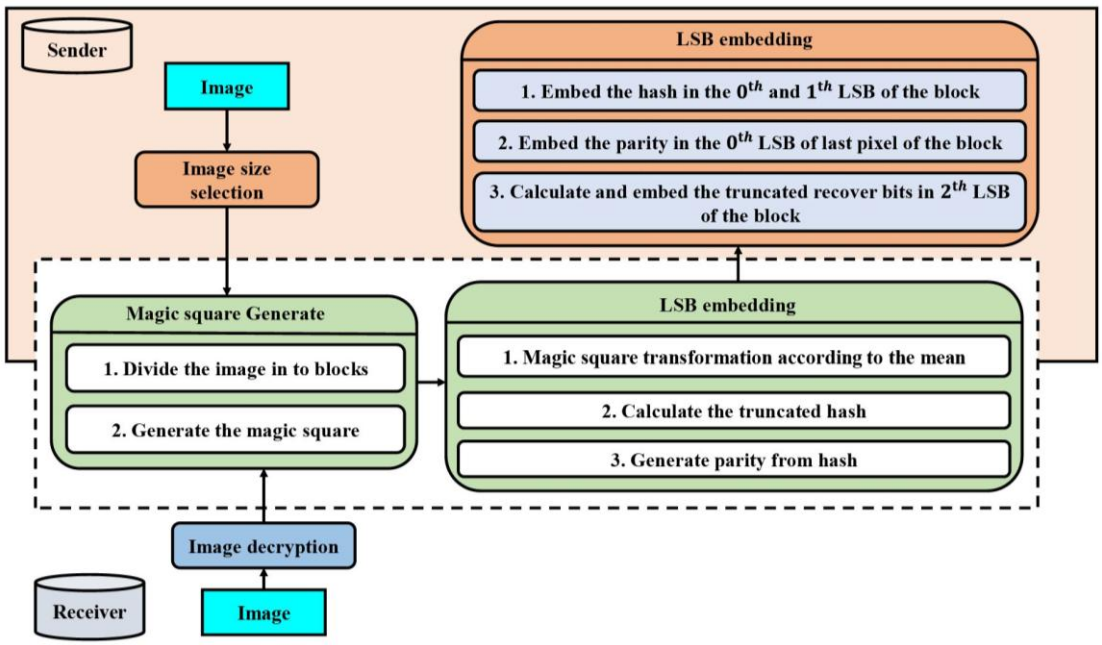


Fig. 1. Architecture of the proposed methodology of EK-ECDH with MS in image encryption.

Decryption uses an identical secret key K to reconstruct the matrix framework; the sums of ciphertext C_1 to C_{10} are considered as the system constants of a 10-linear equation. To use a large matrix, MS5 provides higher complexity and resistance compared to a smaller matrix algorithm. Fig. 1 illustrates the architecture of the proposed methodology of EK-ECDH with MS.

A. Key Generation Using ECC Diffie-Hellman

The ECDH key exchange is the cryptographic approach that allows two parties, such as the sender and receiver, to evaluate a shared secret key through an insecure message by ECC. In this procedure, the sender and receiver select private keys and utilize a scalar multiplication on the determined elliptic curve point based on the output of respective public keys. The public keys are transmitted by connecting the received public key with the private key, while two parties operate with the same shared keys. The ECDH security centers on the complexity of the elliptic curve discrete logarithm issue, ensuring that the unauthorized parties are not allowed to access the secret key from the swapped public keys. The ECDH generates strong security with a small key size compared to conventional approaches. The ECDH needs a small key size, where its security strength is equal to RSA, while the ECDH operates on ECC, and it is a lightweight, smaller key size with insignificant overhead. The EC enables ease of computation due to algebraic addition.

B. Key Generation and Encryption Phase Using Elliptic Key

In the key generation procedure, the pixel of the image is considered as I, private and public keys as d_1 and Q_A is produced, where the set of parameters D is used for encryption schemes that are mentioned as Eq. (2).

$$D = \left(g, FR, x, y, M, u, c\right) \tag{2}$$

where M depicts the key generation point, g is a fixed size, FR is a field representation used in elements F_g, c is a cofactor, and x, y are in a $F_g = E_p$ (a, b) curve.

Collectively, these parameters establish the cryptographic and image-processing domain required for secure encryption and decryption. Entity A follows the steps below. These steps include the derivation of the private key component d_1 through scalar multiplication, which is presented in Eq. (3).

$$d_1 = a \times N_{id} \tag{3}$$

Subsequently, the public key generation in ECC is defined and mentioned in Eq. (4).

$$Q = d_1 P \tag{4}$$

where P represents a base point on the elliptic curve and the private key d_1 is selected randomly from the interval [1, n−1], where n is the order of the base point. A and Q are a public and private key that are shared with the sender and receiver, and the size of the image pixel in the encryption phase. In EK, the encryption mechanism for the image is calculated with P, g, r which are considered prime numbers, e is the public key, and d_2 is the private key, to estimate the encryption mechanism, a mathematical formula is mentioned in Eqs. (5)–(7).

$$l = P \times g \times r \tag{5}$$

$$\varphi(l) = (P-1)(g-1)(r-1) \tag{6}$$

$$1 < e < \varphi(l),\ such\ that\ \gcd(e, \varphi(l)) = 1 \tag{7}$$

The equations specify modulus l as the product of three large prime numbers, P, g, and r. It determines the key space size and directly influences the cryptographic strength of the system. Then $\varphi(l)$ defines Euler's totient function of l, which is mainly useful in key generation, particularly for computing the valid public-private key pairs in modular arithmetic. Further, Eq. (7) defines the selection of the public exponent q. The integer q must lie between 1 and $\varphi(l)$ and be coprime with $\varphi(l)$. This ensures the existence of a modular multiplicative inverse that is required to compute the corresponding private key for secure encryption and decryption. The encryption operation is mentioned in Eq. (8). An encryption message A is applied, while $A < 1$ and the data is encrypted with the public key e and the encrypted message.

$$C = A^e modu \tag{8}$$

Data encryption D_{enc} is transferred while encryption is performed. The shared key is defined in Eq. (9) below.

$$K_s = d_A Q_B = d_B Q_A \tag{9}$$

where the shared key is referred to as K_s and the EK algorithm, the data encryption procedure, and the data are encoded by EK and a unique encryption key. The decryption key is utilized to decrypt an image and confirms the data confidentiality of an image. An EK-ECDH is based on private and public keys, and while this dual organization increases security by keeping private key details confidential and permitting only authorized users to decrypt data, it offers a high level of security.

C. Magic Square

The MS is the procedure to initialize by selecting the previous image to ensure consistent separation in all images. The image size indicates how many pixels are involved in every segment. Dividing an image into pixels and blocks, where each block is a subclass of the entire image, and splitting is necessary to confirm localization on the receiver end. Images are divided into blocks, where a magic square is generated. The MS $N \times N$ matrix is a sum of each column, row, and diagonal, mentioned as a magic constant, while the matrix is utilized to manipulate pixel values within a block, thereby confirming the

modifications in an image are detected later. The created MS is basic for transforming the block data and measuring a hash that makes the difficult component in the verification integrity procedure difficult. By applying the MS transformation to the image, it ensures that any modifications of the image output are significant in various transformed outputs, which aims to encrypt the image. This magic square-based permutation introduces the deterministic yet key-dependent spatial diffusion. It ensures the pixel displacement across color channels while maintaining the low computational overhead. Moreover, the permutation function $\pi : (i, j, c) \to (i', j', c')$ is derived from the magic square structure parameterized by the shared key K_s to ensure invertibility and deterministic decryption. The permutation function is defined based on the key-dependent square matrix S of size $n \times n$. The input image I is divided into non-overlapping blocks of size $n \times n$. For each block, the corresponding magic square S contains unique integrates from 1 to n^2 which determines the permutation order. Let (i, j) denote the spatial position of a pixel within a block. The value $S(i, j)$ represents the new linear position of that pixel after permutation. The block is first vectorized into a one-dimensional array, and pixels are rearranged according to the ascending order of $S(i, j)$.

Specifically, if $S(i, j) = k$, then the pixel at position (i, j, c) is moved to the k-th position in the permuted vector. After permutation, the vector is reshaped back into a 2D block to obtain the new coordinates (i', j'). The channel index c' remains unchanged, i.e., $c' = c$.

Thus, the permutation function π reorders pixel positions within each block using a deterministic, key-dependent mapping derived from the magic square structure.

In the transformation stage, a procedure of three key phases, such as applying the MS transformation, evaluating the truncated hash, and producing a parity bit. Each block is managed based on the MS, which is influenced by the block mean pixel value, which decides that the MS is altered into different modifications of the block. After transforming the block, a truncated hash is created that relies on modified block pixel values and a hash-performed unique identifier for the block, and helps encrypt the image and alterations in the image. The parity bit assists in an integrity check, thereby ensuring that the modifications in the image block are detected. Consequently, to ensure reproducibility, the magic square-based permutation stage is formally defined. The input image is taken as I with the size of $M \times N$, which is divided into non-overlapping blocks of size $B \times B$. Here, B denotes the predefined constant; by default, it is chosen as $B = 8$ to balance computational efficiency and diffusion strength. For each block, a magic square matrix $MS \in \mathbb{Z}^{B \times B}$ is generated using a deterministic key-dependent seed derived from the shared secret. The magic square contains unique integers from 1 to B^2 by defining a permutation order. The scrambling in the magic square is performed by rearranging pixel positions according to the ascending order of magic square values. Particularly, the pixel at position (i, j) is mapped to the k^{th} position in the vectorized block if the value $MS(i, j) = k$. Similarly, if the image dimensions are not divisible by B, then zero-padding is applied to the boundary to form complete blocks. The padding is removed after decryption to restore the original image size. Moreover, the magic square stage performs position permutation only, where the pixel values remain unchanged at this stage, while ensuring confusion without altering intensity distribution. Particularly, this magic square is generated using a key-dependent deterministic algorithm to ensure identical reconstruction during decryption.

In the receiver stage, a similar procedure of MS transformation and generation is employed by the sender. In the receiver stage, the stages are necessary for following a precise transformation, which is applied to every image block in an embedding procedure. Though using the same block size and magic constant values regained from decrypted data, the receiver is recreating an MS and achieves a consistent transformation on each block. This permits a receiver to extract hash values and compare them with expected values, which ensures that the image is encrypted. The transformation of the image ensures consistency between the original and received image, and in the image verification. The pseudo-code for the encryption is given in Algorithm 1.

Algorithm 1: Image Encryption Procedure

Input:
Plain image $I \in \mathbb{R}^{H \times W \times C}$,
Elliptic curve domain parameters (p, a, b, G, n)
Output:
Start:
Encrypted image I_e
Select private keys d_A, $d_B \in [1, n\text{-}1]$.
Compute public keys $Q_A = d_A G$ and $Q_B = d_B G$
Derive a shared secret key using Eq. (9).
Apply the function to K_s to generate a session key K.
Generate a magic square permutation matrix P_{ms} using K
Permute the pixels of I using P_{ms}
Perform diffusion using key-dependent XOR and modular operations.
Obtain the encrypted image I_e.
Stop.

Moreover, the pseudo-code for the decryption is given in Algorithm 2.

Algorithm 2: Image Decryption Procedure

Input:
encrypted image I_e
Elliptic curve domain parameters (p, a, b, G, n)
Output:
Recovered plain image I
Start:
Recompute the shared secret key K_s.
Apply the same hash function to derive the session key K.
Regenerate the magic square permutation matrix P_{ms}.
Reverse the diffusion process using K.
Apply the inverse permutation P_{ms}^{-1}.
Recover the original image I.
Stop.

IV. Experimental Results and Discussion

The experimental results of the proposed EK-ECDH with the MS model are detailed in this section. The simulation of the proposed model is done using MATLAB 2024b software. The system configuration is described below:

- RAM: 16 GB
- Processor: Intel i5
- Operating OS: Windows 10
- GPU: 6 GB
- HDD: 1 TB

All the encryption and decryption times reported are averaged over 10 independent runs to ensure measurement stability. All the experiments are executed under identical hardware and software conditions. Before recording the execution time, an initial warm-up run is performed to mitigate the Just-in-Time (JIT) compilation and the caching effects. The timing measurements are obtained using MATLAB's built-in tic-toc functions, and the mean value across runs is reported.

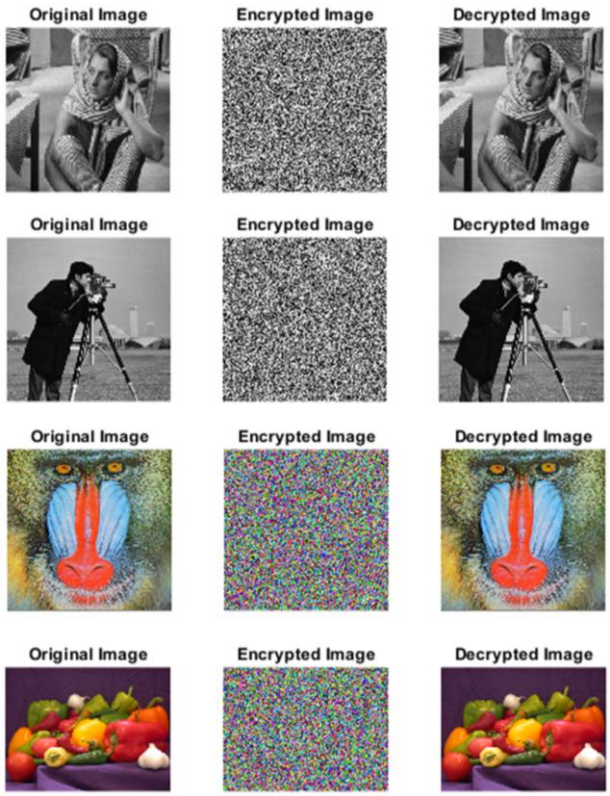


Fig. 2. The original, encrypted, and decrypted images for Barbara, Cameraman, Monkey, and Peppers.

A. Performance Analysis

In the performance analysis section, the various encryption techniques, such as Rivest-Shamir-Adleman (RSA), ECC, Diffie-Hellman Key Exchange (DHKE), and Advanced Encryption Standard (AES), are compared with the proposed EK-ECDH with Magic Square (MS). The objective of performance analysis is to estimate the performance and effectiveness of the proposed EK-ECDH with MS, which increases image encryption. By involving an encryption procedure and using related performance metrics, an experimental output generates insights into the efficiency and security of image encryption applications. The proposed EK-ECDH with MS, where encrypted and decrypted images are attained by publicly available datasets, namely, Barbara, Cameraman, Monkey, and Peppers standard images, is presented in Fig. 2.

In Fig. 2, every image, such as Barbara, Cameraman, Monkey, and Peppers, and the original, encrypted, and decrypted images are presented from left to right. The encrypted image looks like noise-like patterns with no visible structure and signifies that the proposed EK-ECDH with MS efficiently hides the image and resists visual attacks. After decryption, the images recognize the original image while ensuring that the proposed EK-ECDH with MS preserves image quality with minimal distortion, which is reliable with attained high Peak Signal-to-Noise Ratio (PSNR), (SNR), and low level of Mean Squared Error (MSE) values. Moreover, the security of the proposed scheme entirely depends on the elliptic curve parameters' private key size. With the 256-bit private key, the effective key space exceeds 2^{256}, which is computationally not feasible for exhaustive search.

TABLE I. Performance Analysis of Different Encryption Techniques for the Proposed EK-ECDH with MS Using SNR, PSNR, MSE, LED, and NC Metrics

Methods	SNR (dB)	PSNR (dB)	MSE	LSD	NC
RSA	18.5	28.3	72.4	1.92	0.86
ECC	20.8	30.1	58.7	1.65	0.90
DHKE	21.4	31.2	52.9	1.48	0.92
AES	22.6	32.4	46.3	1.32	0.94
Proposed EK-ECDH with MS	25.9	35.7	32.8	1.12	0.98

The performance analysis of the proposed EK-ECDH with the MS model outperforms various encryption methods, such as RSA, ECC, DHKE, and AES, across all evaluation metrics is presented in Table I. The EK-ECDH with MS attains high values for SNR, PSNR, and NC values, showing the high quality and strong correlation with the original image after encryption. The low MSE and LSD values establish minimized distortion, and good structural preservation ensures that the proposed EK-ECDH generates strong security and increases the image quality compared to other existing approaches. Table II presents the analysis of RSA, ECC, AES, DHKE, and the proposed EK-ECDB with MS based on key size, encryption, and decryption time.

Table II depicts that the proposed EK-ECDH with MS model attains low encryption and decryption time across both key sizes of 50 and 100, representing the computational efficiency compared to RSA, ECC, AES, and DHKE. The conventional approaches, such as ECC and RSA, obtain a high level of encryption and decryption time; the proposed EK-ECDH with MS performs better, particularly for small key sizes. The proposed EK-ECDH with MS obtains low encryption and decryption due to the utilization of lightweight key change operations and minimized complexity, which permits images to be encrypted. Table III depicts the performance comparison of standard benchmark images, such as Barbara,

Cameraman, Baboon, and Peppers, with the proposed EK-ECDH with MS.

TABLE II. PERFORMANCE ANALYSIS OF ENCRYPTION AND DECRYPTION TECHNIQUES BASED ON KEY SIZES (BYTES) FOR THE PROPOSED EK-ECDH WITH MS

Method	Key Size (bytes)	Encryption Time (s)	Decryption Time (s)
RSA	50	0.142	0.128
	100	0.231	0.214
ECC	50	0.096	0.082
	100	0.158	0.143
DHKE	50	0.088	0.077
	100	0.149	0.131
AES	50	0.054	0.048
	100	0.064	0.054
Proposed EK-ECDH with MS	50	0.041	0.036
	100	0.072	0.066

Table III results indicate that the proposed EK-ECDH achieves high performance across all the standard benchmark images, such as Barbara, Cameraman, Baboon, and Peppers, that are compared with RSA, ECC, DHKE, and AES, which illustrates better preservation of the image after the decryption procedure. The proposed EK-ECDH model attains high values due to the key exchange mechanism and reduces the complexity during the encryption and decryption process. The proposed EK-ECDH model has the capability to generate better security with minimal degradation, which allows rebuilding the image quality. Table IV presents the computational complexity analysis of the proposed EK-ECDH for various metrics.

TABLE III. PERFORMANCE ANALYSIS OF RSA, ECC, DHKE, AES, AND PROPOSED EK-ECDH WITH MS ACROSS STANDARD BENCHMARK IMAGES

Methods	Barbara	Cameraman	Baboon	Peppers
RSA	0.86	0.88	0.84	0.87
ECC	0.90	0.91	0.88	0.92
DHKE	0.92	0.93	0.90	0.94
AES	0.94	0.95	0.93	0.96
Proposed EK-ECDH	0.98	0.96	0.97	0.98

TABLE IV. PERFORMANCE ANALYSIS OF COMPUTATIONAL COMPLEXITY FOR THE PROPOSED EK-ECDH WITH MS

Methods	Training Time (s)	Inference Time (s)	Execution Time (s)	Key Generation Time (s)	Memory Usage (KB)
RSA	1.92	0.134	0.268	0.152	984
ECC	1.41	0.102	0.204	0.098	812
DHKE	1.33	0.096	0.188	0.085	772
AES	0.98	0.064	0.128	0.054	650
Proposed EK-ECDH with MS	0.72	0.049	0.096	0.031	528

The proposed EK-ECDH model achieves low levels of training, inference, execution, and key generation times, and less memory usage. The proposed EK-ECDH requires less storage memory compared to conventional approaches, while the proposed model highlights the efficiency of the proposed EK-ECDH to deliver efficient image encryption. The comparison with these models, including RSA, ECC, DHKE, and AES, is not intended to suggest direct functional equivalence because these algorithms work at different abstraction levels. Rather, the evaluation highlights the suitability of the proposed model for image data, where pixel-level permutation combined with lightweight key exchange obtains improved statistical security and computational efficiency when compared to applying the conventional cryptographic primitives directly to image streams. The elliptic curve-based key exchange minimizes redundant computation, thereby reducing the processing and memory access overhead. The proposed EK-ECDH is lightweight and significantly minimizes the computational complexity, which is able to handle strong security for image encryption. Table V illustrates the ablation study of the proposed EK-ECDH across various metrics.

Table V determines the proposed EK-ECDH model ablation study. The integration of the proposed EK-ECDH model performs better because the EK with elliptic curve-based key exchange increases the security and minimizes the distortion during the encryption procedure. The EK generates effective lightweight encryption, ECC distributes strong key generation security, and DH permits the secure key exchange. The associated model reduces pixel-level distortion, minimizes noise propagation, and handles better correlation with the original image. The combination attains low MSE, Log-Spectral Distance (LSD), and enhanced SNR, NC, and PSNR, which permit the proposed model to preserve image quality efficiently compared to the individual components. Fig. 3 demonstrates the original, encrypted, and decrypted images of the histogram and correlation of the original image of Barbara in horizontal, vertical, and diagonal directions.

TABLE V. ABLATION STUDY OF THE PROPOSED EK-ECDH MODEL USING DIFFERENT PERFORMANCE METRICS

Methods	SNR (dB)	PSNR (dB)	MSE	LSD	NC
EK	23.5	33.1	40.2	1.25	0.95
ECC + DH	22.3	31.8	50.4	1.45	0.92
EK + DH	24.6	34.2	36.7	1.15	0.96
ECC	20.8	30.1	58.7	1.65	0.91
Proposed EK-ECDH with MS	25.9	35.7	32.8	1.12	0.98

Fig. 3 illustrates that the original Barbara image is a raw input utilized to estimate the encryption and decryption performance, which is the collection of visual details, textures, and data of pixels. In the encryption version, the image looks distorted, and noise denotes that the encryption procedure efficiently obscures visual content while no recognizable construction is visible, thereby ensuring a high level of security. The decrypted image stores the original image with high quality and minimal alteration. The proposed EK-ECDH model ensures that the model accurately recovers the original image. The horizontal graph exhibits the distribution of pixel intensity

values across every row of the image, and the vertical graph illustrates a pixel intensity distribution across the column. While the encryption projection is irregular, the original and decrypted estimates align closely. The histogram of the original image is the differentiation of grayscale value in the Barbara image; the encrypted image of the histogram shows no correlation or meaningful pattern with the original image. A decrypted histogram is matched with an original histogram image, ensuring the recovery of the image and minimal information loss after the encryption and decryption procedure. Fig. 4 represents the correlation of the original image for the Cameraman in horizontal, vertical, diagonal, and histogram analyses of the original, encrypted, and decrypted images of the Cameraman. Here, the low correlation values demonstrate that the encryption process effectively removes inherent spatial redundancy.

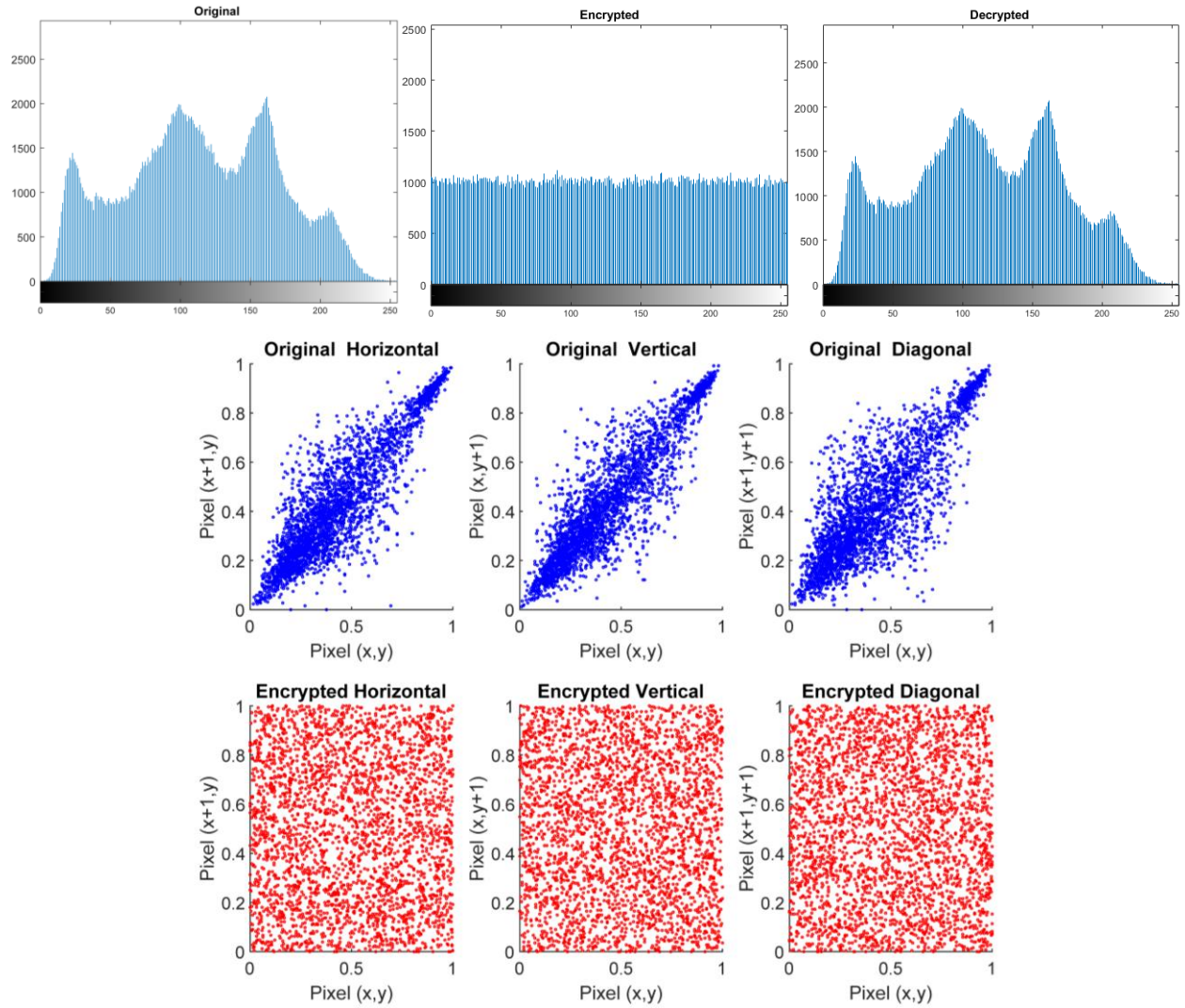
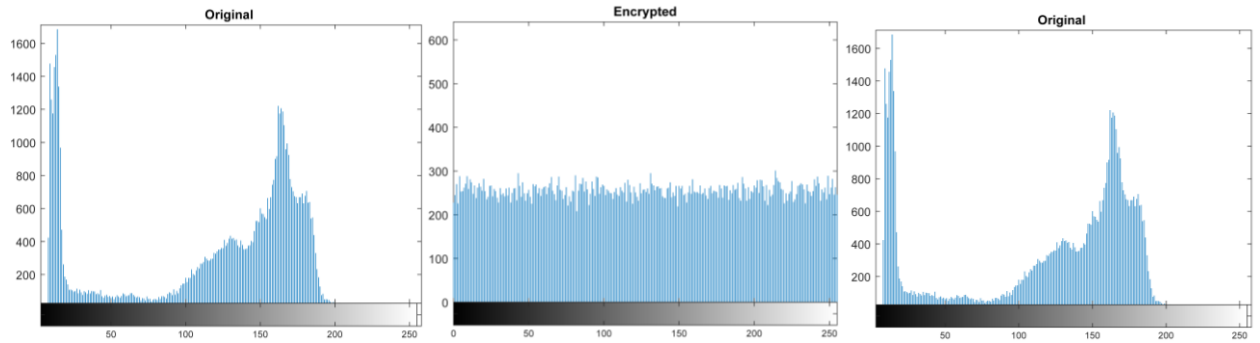


Fig. 3. Correlation of the original image for Barbara in horizontal, vertical, diagonal, and histogram analysis of original, encrypted, and decrypted images of Barbara.



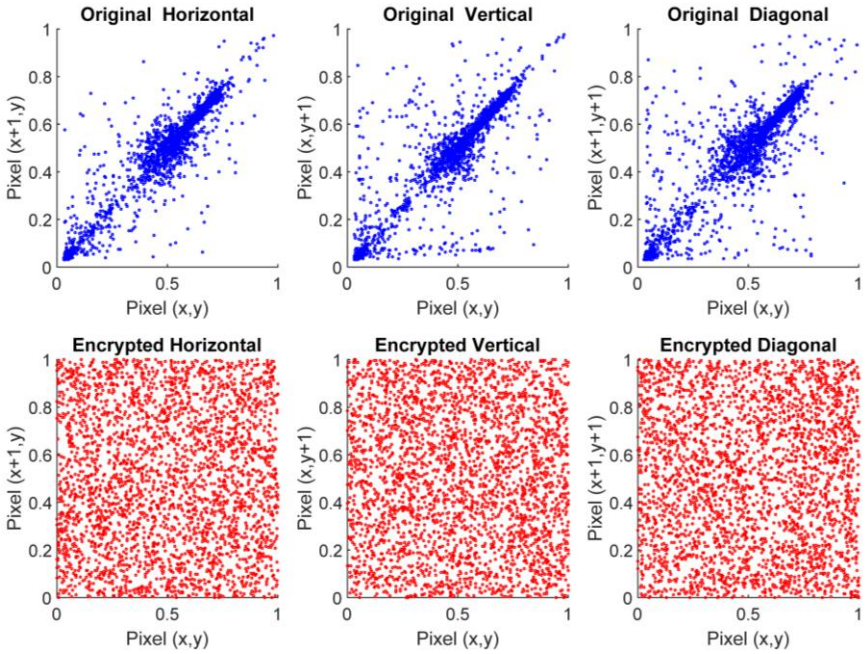


Fig. 4. Correlation of the original image for the Cameraman in horizontal, vertical, diagonal, and histogram analysis of original, encrypted, and decrypted images of the Cameraman.

Moreover, the resolution scalability is analyzed to evaluate the scalability of large-scale images. The proposed encryption scheme is tested on increasing image resolution while preserving the identical encryption parameters. The standard benchmark images are resized to 256 × 256, 512 × 512, and 1024 × 1024 resolutions. Moreover, the block size B remains constant across all tested resolutions, and it ensures that the performance variation reflects the scalability rather than structural modification. For each resolution, the encryption time, Number of Pixels Change Rate (NPCR), Unified Average Changing Intensity (UACI), and Shannon entropy are measured to evaluate the computational scalability and security consistency, which is given in Table VI.

TABLE VI. EVALUATES THE SCALABILITY OF THE PROPOSED MODEL WITH HIGH-RESOLUTION IMAGES FROM THE STANDARD BENCHMARK DATASET

Resolution	Time (s)	NPCR (%)	UACI (%)	Entropy
256 × 256	0.08	99.61	33.42	7.998
512 × 512	0.32	99.60	33.38	7.997
1024 × 1024	1.28	99.58	33.35	7.996

From Table VI, the results indicate that encryption time increases proportionally with image resolution, confirming linear computational scalability. Particularly, the NPCR, UACI, and entropy values remain stable across all the resolutions. It demonstrates the security properties of the proposed scheme that are independent of image size and remain effective for high-resolution images.

TABLE VII. ARNOLD'S CAT MAP WITH ECC AND GA [17] IS COMPARED WITH THE PROPOSED EK-ECDH WITH MS ACROSS DIFFERENT STANDARD IMAGES, SUCH AS CAMERAMAN, BABOON, PEPPERS, AND BARBARA

Images	Methods	Entropy	NPCR	UACI
Barbara	Arnold's cat map with ECC and GA [17]	7.9987	0.9958	0.318
	Proposed EK-ECDH with MS	8.2145	0.9961	0.425
Cameraman	Arnold's cat map with ECC and GA	7.9987	0.9958	0.3365
	Proposed EK-ECDH with MS	8.2145	0.9962	0.3781
Baboon	Arnold's cat map with ECC and GA [17]	7.9987	0.9958	0.336
	Proposed EK-ECDH with MS	8.0154	0.9962	0.345
Peppers	Arnold's cat map with ECC and GA [17]	7.9986	0.9958	0.3340
	Proposed EK-ECDH with MS	8.1572	0.9964	0.3514

TABLE VIII. COMPARATIVE ANALYSIS OF PROPOSED EK-ECDH WITH MS IS COMPARED WITH ARNOLD'S CAT MAP WITH ECC AND GA [17]

Methods	Entropy	Correlation Coefficient	Barbara	NPCR	UACI
Arnold's cat map with ECC and GA [17]	7.9987	Horizontal	−0.0045	99.67	33.49
		Vertical	−0.0070		
		Diagonal	0.0008		
Proposed EK-ECDH with MS	7.9992	Horizontal	−0.0024	99.71	35.67
		Vertical	−0.0065		
		Diagonal	0.0005		

B. Comparative Analysis

In this section, the proposed EK-ECDH with MS model is compared with existing approaches, such as Arnold's cat map with ECC and GA [13], LPWLCM [15], and the enhanced Royat model with ECC [16]. The performance metrics, such as entropy, NPCR, UACI, key size (bits), encryption, and decryption times, are considered to evaluate and estimate the proposed EK-ECDH with MS model performance. Table VII exhibits the comparative analysis of the proposed EK-ECDH with MS using various standard images, namely, cameraman, baboon, peppers, and Barbara, with Arnold's cat map with ECC and GA [17]. Table VIII depicts the comparative analysis of the proposed EK-ECDH with MS and the comparison with Arnold's cat map with ECC and GA [17] for analyzing entropy, correlation coefficient, NPCR, and UACI. The LPWLCM [19] model is compared with the proposed EK-ECDH with MS for analyzing the model through NPCR, UACI, and execution time of encryption and decryption procedures, which are presented in Table IX. Table X illustrates the comparative analysis of the enhanced Royat model with the ECC [20] model, compared with the proposed EK-ECDH with MS.

TABLE IX. ILLUSTRATES THE COMPARATIVE ANALYSIS OF THE PROPOSED EK-ECDH COMPARED WITH LPWLCM [19]

Images	Methods	Entropy	NPCR	UACI	Execution Time in the Encryption Process	Execution Time in the Decryption Process
Barbara	LPWLCM [19]	7.9987	0.9958	0.318	NA	NA
	Proposed EK-ECDH with MS	8.0124	0.9962	0.345	0.654	0.457
Cameraman	LPWLCM [19]	7.9987	0.9958	0.3365	0.7644	0.3614
	Proposed EK-ECDH with MS	8.574	0.9965	0.3472	0.6234	0.3421
Baboon	LPWLCM [19]	7.9987	0.9958	0.336	1.4734	1.29797
	Proposed EK-ECDH with MS	8.0146	0.9931	0.357	1.2455	1.1843
Peppers	LPWLCM [19]	7.9986	0.9958	0.3340	1.5206	1.29316
	Proposed EK-ECDH with MS	8.1528	0.9965	0.3457	1.347	1.2245

TABLE X. THE ENHANCED ROYAT MODEL WITH ECC [20] IS COMPARED WITH THE PROPOSED EK-ECDH WITH MS USING KEY SIZE, ENTROPY, ENCRYPTION, AND DECRYPTION TIME

Methods	Key Size (Bits)	Entropy for cipher image	Encryption Time (s)	Decryption time (s)
Enhanced Royat model with ECC [20]	512	7.9992	0.35	0.28
Proposed EK-ECDH with MS	436	8.3245	0.24	0.17

Table VII presents that the proposed method is compared with Arnold's cat map with ECC and GA, in that entropy values show strong randomness, while the proposed EK-ECDH with MS attains high entropy and enhanced NPCR across all images, thereby reflecting high pixel randomness. Table VIII estimates the correlation coefficient in three directions, such as H, V, and D. A lower correlation between adjacent pixels is necessary for secure image encryption. The proposed EK-ECDH with MS signifies the correlation coefficients near zero in all directions, which illustrates that there is no statistical relationship among adjacent encrypted pixels. The NPCR and UACI metrics are computed for evaluating the cryptographic security. Where the higher NPCR and UACI values indicate strong diffusion while ensuring that small changes in the plaintext propagate unpredictably throughout the ciphertext. Moreover, a one-bit variation in the encryption key produces a completely different ciphertext image that confirms the strong key sensitivity. In addition, the entropy values approaching the ideal value indicate that ciphertext pixels are uniformly distributed while reducing predictability. Table IX compares the proposed EK-ECDH with MS and handles similar and better entropy, which executes faster in both encryption and decryption, particularly in large images. The values referring to Not Available (NA) demonstrates that a specific value is not presented in the respective work. The enhanced Royal with ECC utilizes 512-bit keys (Table X); the proposed EK-ECDH with MS attains a smaller and more effective key size, indicating better security.

C. Discussion

The advantages of the proposed EK-ECDH with MS and the limitations of existing approaches are discussed in this section. The existing approaches exhibit limitations such as computational overhead, high resolution of video, and modification of image pixels, which are considered drawbacks of the existing model. The computational overhead affects the model resource, and it does not perform well in ultrasound images; image pixel modifications reduce the encryption speed, which affects the entire encrypted image. Each limitation leads to a reduction in the model's efficiency for image encryption. The EK-ECDH with MS is proposed to overcome these limitations; the proposed EK-ECDH with MS image encryption effectively manages color and high-resolution images while generating strong security and minimizing pixel correlation. The EK-ECDH produces strong secret keys by ECDH, ensuring high non-linear mixing of pixel values. The proposed EK-ECDH requires lower storage memory compared to conventional approaches, while the proposed model highlights the efficiency of the proposed EK-ECDH in delivering efficient image encryption. The elliptic curve-based key exchange minimizes redundant computation, thereby reducing the processing and memory access overhead.

V. CONCLUSION

In this manuscript, the proposed EK-ECDH with MS was utilized to enhance the image quality with encryption, which efficiently handled the color and high-resolution

image while it produced a higher level of security. The EK was produced by ECHD and magic square-based pixels, where only authorized users could encrypt or decrypt the image. By modifying pixel values and positions, it was difficult to encrypt the image with the correct decryption key. The ECC generated the lightweight key framework for producing and securely transferring encryption keys with high security in small key sizes. By associating ECDH with magic square-based pixel permutation, the proposed EK-ECDH model attained robust key management and efficient pixel-level scrambling, thereby ensuring image security. In the experimental results, the standard images such as Cameraman, monkey, barbara, and peppers were utilized to evaluate and estimate the model's performance.

APPENDIX A NOTATION TABLE

Notation	Definition
$E_p(a, b)$	Elliptic curve over a finite field
G	Base point
d_A, d_B	Private keys
Q_A, Q_B	Public keys
K_s	Shared secret key
I, I_e	Original and encrypted image
P_{ms}	Magic square permutation
K_{11}	Key-dependent parameter
K_{222}	Diffusion coefficient
K_{12}	y-coordinate of the elliptic curve point K_1
K_{21}	x-coordinate of the elliptic curve point K_2

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

BV: Created the overall system architecture, conceived the research problem, and oversaw the implementation process. Oversaw the manuscript's preparation and made sure all technical details were correct. KM: Developed the fundamental implementation modules, designed and coded algorithms, carried out experimental testing, and helped to analyze the data and understand the findings that were used in the paper. VUD: Performed performance evaluations, prepared datasets, and helped implement auxiliary modules. Helped with the writing of the manuscript and added to the explanation of the results. SKC: Carried out the literature review, validated the implemented system, and cross-verified experiments. Responsible for proofreading, formatting according to journal guidelines, and final review of the paper. All authors had approved the final version.

ACKNOWLEDGMENT

The authors would like to thank all who are directly or indirectly involved, advised, and coordinated in collecting data and software information for the support of this paper.

REFERENCES

[1] T. Mendis, F. Kandah, and L. Medury, "MATADOR: A magic matrix-based framework for tamper detection and image recovery," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 37, no. 5, 77, 2025.

[2] H. Chen, E. Bai, X. Jiang, and Y. Wu, "A fast image encryption algorithm based on improved 6-D hyper-chaotic system," *IEEE Access*, vol. 10, pp. 116031–116044, 2022.

[3] R. I. Abdelfatah, "Quantum image encryption using a Self-Adaptive Hash Function-Controlled Chaotic Map (SAHF-CCM)," *IEEE Access*, vol. 10, pp. 107152–107169, 2022.

[4] X. Zhang and M. Liu, "Multiple-image encryption algorithm based on the stereo zigzag transformation," *Multimedia Tools Appl.*, vol. 83, no. 8, pp. 22701–22726, 2023.

[5] 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," *Multimedia Tools Appl.*, vol. 84, no. 13, pp. 11503–11531, 2024.

[6] N. Sumalatha and K. Baskaran, "An efficient multiuser threshold-based dynamic cipher puzzle with secure key generation for DoS attacks in smart grid applications," *Trans. Emerg. Telecommun. Technol.*, vol. 36, no. 7, e70204, 2025.

[7] Y. Hui, H. Liu, and P. Fang, "A DNA image encryption based on a new hyperchaotic system," *Multimedia Tools Appl.*, vol. 82, no. 14, pp. 21983–22007, 2021.

[8] Z. Man, C. Gao, Y. Dai, and X. Meng, "Dynamic rotation medical image encryption scheme based on improved Lorenz chaos," *Nonlinear Dyn.*, vol. 112, no. 15, pp. 13571–13597, 2024.

[9] M. M. Wang, X. G. Song, S. H. Liu, X. Q. Zhao, and N. R. Zhou, "A novel 2D log-logistic–sine chaotic map for image encryption," *Nonlinear Dyn.*, vol. 113, no. 3, pp. 2867–2896, 2024.

[10] D. Raju, L. Eleswarapu, M. S. Pranav, and R. K. Sinha, "Multi-level image security using elliptic curve and magic matrix with advanced encryption standard," *Multimedia Tools Appl.*, vol. 81, no. 26, pp. 37783–37803, 2022.

[11] D. R. Ibrahim, R. Abdullah, and J. S. Teh, "An enhanced color visual cryptography scheme based on the binary dragonfly algorithm," *Int. J. Comput. Appl.*, vol. 44, no. 7, pp. 623–632, 2022.

[12] F. Musanna, D. Dangwal, and S. Kumar, "Novel image encryption algorithm using fractional chaos and cellular neural network," *J. Ambient Intell. Humanized Comput.*, vol. 13, no. 4, pp. 2205–2226, 2021.

[13] S. Gupta, A. Nayyar, and Nitish, "Enhancing color image encryption scheme with a chaotic map and Fibonacci transform," *Cluster Comput.*, vol. 28, no. 15, 993, 2025.

[14] H. Wen and Y. Lin, "Cryptanalysis of an image encryption algorithm using quantum chaotic map and DNA coding," *Expert Syst. Appl.*, vol. 237, 121514, 2024.

[15] J. Jackson and R. Perumal, "A robust image encryption technique based on an improved fractional order chaotic map," *Nonlinear Dyn.*, vol. 113, no. 7, pp. 7277–7296, 2024.

[16] E. Güvenoğlu, "An image encryption algorithm based on multi-layered chaotic maps and its security analysis," *Connection Sci.*, vol. 36, no. 1, 2312108, 2024.

[17] S. Kumar and D. Sharma, "A chaotic-based image encryption scheme using elliptic curve cryptography and genetic algorithm," *Artif. Intell. Rev.*, vol. 57, no. 4, 87, 2024.

[18] R. Yadav and P. Singh, "Security enhancement of three POMs-based interference algorithm using elliptic curve cryptography," *J. Opt.*, pp. 1–18, 2025.

[19] K. Pandey and D. Sharma, "Novel image encryption algorithm utilizing hybrid chaotic maps and elliptic curve cryptography with genetic algorithm," *J. Inf. Secur. Appl.*, vol. 89, 103995, 2025.

[20] K. M. Singh, L. D. Singh, and T. Tuithung, "Improvement of image transmission using chaotic system and elliptic curve cryptography," *Multimedia Tools Appl.*, vol. 82, no. 1, pp. 1149–1170, 2022.

[21] N. Rani, V. Mishra, and S. R. Sharma, "Image encryption model based on novel magic square with differential encoding and chaotic map," *Nonlinear Dyn.*, vol. 111, no. 3, pp. 2869–2893, 2022.

[22] P. L. Sharma, S. Gupta, A. Nayyar, M. Harish, K. Gupta, and A. K. Sharma, "ECC-based novel color image encryption methodology using primitive polynomial," *Multimedia Tools Appl.*, vol. 83, no. 31, pp. 76301–76340, 2024.

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).