Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 7, No. 4, October 2025, pp: 1178-1198 **e-ISSN: 2656-8632**

RESEARCH ARTICLE OPEN ACCESS

Secure Image Transmission using Quantum-Resilient and Gate Network for Latent-Key Generation

Malige Gangappa[1], Balla V V Satyanarayana[1], and Dheeraj A[2]

[1] Department of CSE, VNR VJIET, JNTUH University, Hyderabad, India
[2] Department of CSE, University of North Texas, Texas, USA

Corresponding author: Balla V V Satyanarayana. (e-mail: 23071d5801@vnrvjiet.in), **Author(s) Email**: Malige Gangappa (e-mail: gangappa_m@vnrvjiet.in, Dheeraj A (e-mail: dheerajavadhanula@gmail.com)

Abstract Recently, deep learning-based techniques have undergone rapid development, yielding promising results in various fields. For making more complex operations in day-to-day tasks, the arbitrary resolution of JPEG image data security requires more than just deep learning in this modern era. To overcome this, our research introduces a pioneering synergistic framework for a quantum-resistant deep learning technique, which is expected to provide next-generation robust security in the dynamic resolution of multi-JPEG-image-based joint compression-encryption. Our proposed framework features dual-parallel processing of a dynamic gate network, utilizing a convolutional neural network for specialization detailing and quantum-inspired transformations. These transformations leverage Riemann zeta functions for depth feature extraction, integrated with a chaotic sequence and dynamic iterations to generate a latent-fused chaotic key for image joint compression and encryption. Further, the authenticity of an encrypted image that is bound by a secure pattern derived from a random transform variance anchors cryptographic operations. Then, bound data transmitted through a Synergic Curve Key Exchange Engine fused with renowned Chen attractors to generate non-invertible keys for transmission. Finally, experimental results of the image reconstruction quality measured by the structural similarity index metric were 98.82$\pm$1.12. Security validation incorporates different metrics by addressing the entropy analysis to quantify resistance against differential and statistical attacks, with a yield of 7.9980$\pm$0.0015. In conclusion, the whole implementation uniquely combines latent-fused chaotic with improved key space analysis 2^{640} for discrete cosine transform quantization with authenticated encryption, establishing an adversarial-resistant pipeline that simultaneously compresses data and validates integrity through pattern-bound authentication.

Keywords Quantum-Resistant Image Processing, Dynamic Gate Network Encoder, Dual-Curve Key Exchange, Secure Pattern, Latent-Fused Chaotic Key.

I. Introduction

As technology advances, more individuals use networks to send and store photographs and other data. When delivering sensitive material, it's often desired that only the intended receiver may view it. Solving security issues during information transfer is a significant task. Encrypting images is crucial for maintaining their security and secrecy [1]. Many researchers have introduced various designs with different technological domains to withstand different attacks or data theft. Before securing the image, we need to understand which type of image format should be evaluated. A popular digital picture format created for the effective storing and transfer of photographic images is the Joint Photographic Experts Group (JPEG) [2]. It is the default standard for photographs on the web, digital cameras, and smartphones due to its ability to strike a balance between image quality and file size [3]. The JPEG format employs a lossy compression technique [4][5], which intentionally discards some image data (such as metadata) to minimize file size while maintaining visual quality for the majority of users, and supports up to 24-bit color [6]. The discrete cosine transform (DCT), which divides an image into frequency components, is the foundation of JPEG compression. JPEG's signature compression efficiency is attained by quantizing and deleting high-frequency data. Despite being lossy, this method prioritizes the information most visible to the human eye [7][8], producing images that are aesthetically pleasing for most uses. Each stage contributes [9] to the final compressed file, including color space conversion (usually to YCbCr), DCT application, quantization, and entropy coding. Further, to improve

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

the compression ratio [10][11][12] with minimal loss of quality by using machine learning and deep learning to overcome traditional limitations and steps towards base-level security via compression. However, that base-level security does not withstand classical computing attacks alone.

Therefore, image encryption [13][14][15] using cryptographic operations has come into the era to provide image data protection. To secure JPEG [16] pictures, they may become illegible if they are broken, or else pixel loss will occur due to direct encryption. To solve this [17][18][19][20], many researchers have worked on compress-then-encrypt, which processes the standard JPEG algorithm utilized before encryption because the unaltered outcome [21][22] can be easily compressed. The encryption here utilizes classical information theory and the separation of cryptography. Therefore, the security [23][24] depends solely on the encryption, as the compression is public. But the compression ratio is high since the compression applies prior to the entropy loss encryption. In the aspect of the attack, resistance is limited to the cryptoanalysis of the encrypted cipher data.

As the years go by, other studies have solved this problem via a joint compression and encryption approach [25][26], which is performed in a single process through the JPEG pipeline and modified or combined to allow encryption steps to occur during compression. The joint approaches, such as the transform domain methods (DCT, DWT, SPIHT), chaos-based scrambling, Chinese remainder theorem (CRT), and orthogonal transforms integrating with machine learning and deep learning domains for improving [27] key space, entropy, peak-to-noise ratio (PSNR), statistical attack, and differential attack resistance, are modeled via Shannon information theory and combinatorial analysis. Additionally, compression technologies that integrate with the standard JPEG pipeline, sometimes with minor block permutations or quantization changes for compression, act as an additional security layer, but the computational overhead and encryption time have increased linearly.

To solve this, studies utilizing selective encryption [28] applied over the process improved joint compress-encrypt techniques with hybridization of automation models (such as machine learning, deep learning, or both) have been developed to reduce the image quality loss, which enables the encryption of some areas of a picture, such as faces or private content, while preserving the other portion of the image in its original, visible condition. To guarantee format compatibility and perceptual security, these techniques [29] frequently work with the DCT coefficients or employ chaos-based algorithms. The objective is to strike a compromise between usability (keeping the file readable by ordinary software) and security (obscuring sensitive material). Therefore, it reduces the encryption time and computational cost. However, the world of computing could be transitioning from binary to the quantum computing era, which can significantly reduce the time required to break the above approaches because they are not robust enough to defend against post-quantum attacks, are deterministic, and lack sufficient entropy sources for cryptographic security. So, the researchers are looking forward to quantum field utilization in image security.

Quantum theory, which encompasses quantum physics and information science, is utilized to process information through computation. Quantum computing-based image encryption [30] has been successfully implemented at both the experimental and prototype levels in specialized hardware chips and academic research environments. However, widespread practical deployment [31] for enterprise-scale or users' image security remains limited due to tiny image pixels (e.g., $\leq 16\times16$), as larger images exceed present qubit, error correction capabilities, and are more costly to afford. To overcome this issue, modern computing is used to simulate quantum theory principles [32], which have been mapped to deep learning algorithms to enhance image security and mitigate the current limitations of quantum hardware. Therefore, the existing mechanisms (quantum principles) [33][34][35] primarily focus on quantum-based feature extraction, quantum-based key generation, and quantum-inspired superposition entanglement, with improved joint approaches to protect against post-quantum attacks and enhance key space analysis. But these approaches are limited to non-arbitrary-sized image resolutions input, single-level processing key features, may leak histogram information due to not encrypting pixel values, and end-to-end latency for quantum-based key exchange delays happening in previous methods.

Therefore, to solve these problems, this study aims to implement the dynamic resolution JPEG multi-image-based joint compression-encryption via a seed map of quantum-inspired deep learning to extract a fusion feature set from the input image (such as day-to-day life JPEG) of arbitrary-sized resolutions for a latent-fused key generation with a minimal computational usage per image via ROI (Region of Interest) approach to our model in compression cycle for optimal resource management and reduce the time consumption, better peak signal-to-noise (PSNR) and structural similarity index metric (SSIM) , and provides resistance to post-quantum attacks of encrypted image via improved key space. However, our preliminary research has been conducted on the findings from various segments of existing research papers.

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

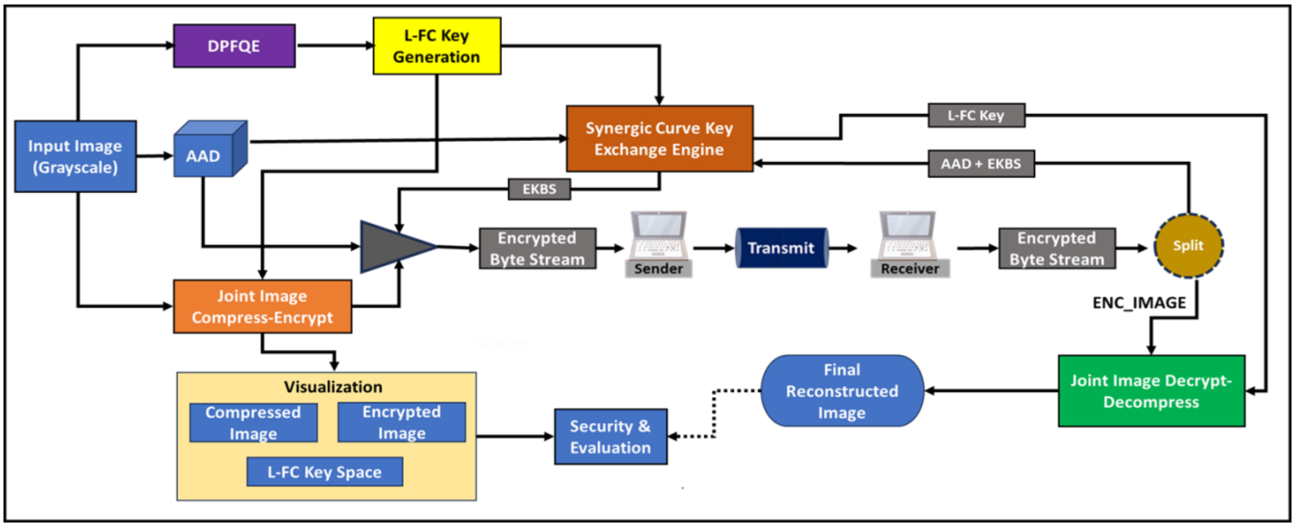

Fig. 1. Synergic Secure Image Processing and Transmission Framework for Arbitrary Resolution JPEG.

This section provides an overview of recent works in individual research areas, including compression, encryption, quantum image processing, and combinational modes. The significant contributions and innovations of our work are: 1) The fully automated model is developed by applying quantum-inspired transformation (quantum principles) and the deep learning network, which could be an early undertaking of strict data lineage to the latent-fused key generation from image content for joint compression encryption, 2) To prevent algebraic analysis, we built a quantum resistance block to transform the latent feature vector nature to complex frequency domains, and the system creates non-linear feature distortions resistant to inversion attacks, 3) The latent-fused chaotic key is generated using prime-based Chinese Remainder Theorem unique solution processes, which latent vectors through iterative prime sieving and residual locking for improving key space, 4) For Secure transmission of a latent-fused chaotic key, dual elliptical curve integration with a pattern-bound key derivative function establishes cross-protocol verification and balance the end-to-end key exchange latency, 5) To reduce computation load and improve reconstruction quality per image using the selective region-based compression approach without degrading encryption.

The rest of this paper is organized in sections as follows: Section II represents an overview and implementation of our proposed framework for joint dynamic resolution JPEG-image-based compression encryption. Section III describes the evaluation and analysis of our framework's process quality and security metrics. Section IV presents a discussion on the efficiency and performance comparison with other existing methods, and Section V provides a concise conclusion and outlines the future follow-up of this paper.

II. Method

Our framework represents a novel synergic secure image processing and transmission model, that was created with quantum computing principles, chaos dynamics, and deep learning techniques as shown in Fig. 1. As its core component, the technology transforms visual JPEG image data into compressed blocks followed by encrypted with inter-dependent three-layered format of unique keys derivation from the input image’s own features map, making the image resistant to both cryptanalysis including post-quantum and exploiting JPEG format attacks. Furthermore, the following sections provide a detailed description of our model's foundations and operational flow.

A. Dataset JPEG Ingestion and Preprocessing

For real-time analysis perception, we downloaded freely available JPEG images from our machine’s OMEN wallpaper application to evaluate the model. We can also use open-source image datasets or day-to-day images. However, we utilized OMEN wallpaper images with arbitrary resolution as input to an ensemble meta classifier model [36], which yielded stego-free JPEG images as output. The main aim of using a meta classifier of images for defending the attacks exploiting using JPEG format and processing vulnerabilities (such as steganography-based, metadata attacks, etc.) to any security-based model to minimize the framework's working principles' exploits. Furthermore, the obtained stego-free images are input into our image preprocessing module, which is part of the entire model, to process and evaluate samples, as shown in Fig. 2. As part of the image preprocessing for

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

our model, it accepts different resolutions with batch processing of JPEG images for loading and processing.

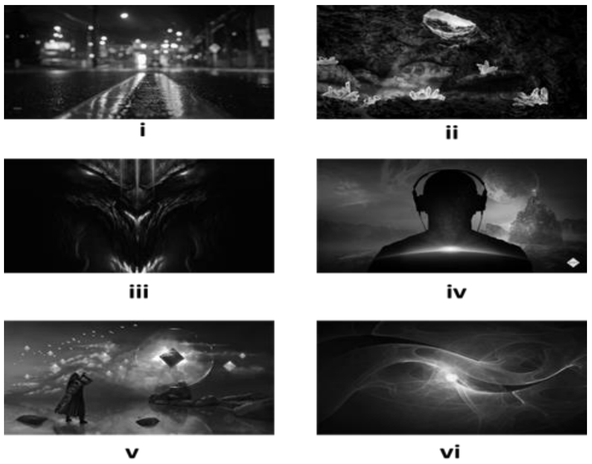

Fig. 3. i) Carretera, ii) Crystals, iii) Diablo, iv) Reveries, v) Sky, and vi) Spark are some samples for the working process.

For our understanding, we consider a single JPEG image of arbitrary resolution (m*n), where m and n are any resolution that support both square and rectangular formats. After loading the JPEG image for standardized processing, if the input image is grayscale, then directly process it to the next step; else convert it to single-channel grayscale $I_{gray} \epsilon [0,255]^{m*n}$ using OpenCV's library import of IMREAD_GRAYSCALE function as shown in Eq. (1) and preserve the original input dimensions for the reconstruction phase as metadata in byte form.

$$I_{gray}(i,j) = 0.299RED(i,j) + 0.586GREEN(i,j) + 0.114BLUE(i,j) \quad (1)$$

where i, j are pixel positions in range of 0≤i<m, 0≤j<n, and RED, GREEN, and BLUE are the color channels.

$$I_{resized} = resize(I_{gray}, 64 \times 64) \quad (2)$$

Those I_{gray} are resized $(I_{resized})$ to a fixed dimension as shown in Eq. (2). This process standardization ensures compatibility with our Dual Process Fusion Quantum Encoder (DPFQE) for input requirements and process represented in Section II.B. Note that pixel values are normalized to the range of [0,1], converting the input JPEG image into a tensor format suitable for neural network processing.

B. Dual Process Quantum Encoder for Feature Fusion Extraction

From Section II.A, the results of the pre-processed images were input into our DPFQE, which is a custom classical deep learning model with a dynamic gate and quantum-inspired processing as shown in Fig. 3. The DPFQE employs two parallel processing pathways to extract feature dimensions. The pathways are named Dynamic Weighting Gate Network Block (DWGNB) and Quantum Resistant Block (QRB). For our understanding of the flow, we first analyze the DWGNB using its process flow of Algorithm 1. The process starts with the gate network method, which is a small CNN that includes of a Conv2D (Convolutional 2D) layer, a flattened layer, a dense layer, and an output layer to generate dynamic weights per image for classical processing. Here, the Conv2D layer consists of 8 filters with a stride assignment. This layer aggressively down-samples the image and utilizes the Rectified Linear Unit (ReLU) activation function. Then, flatten the output of Conv2D to a 1D vector of 512 values and connect a fully connected layer with 32-neuron units (hidden layer) by utilizing the activation function of ReLU, finally, for producing three weights that are positive and sum to 1 via an output layer with

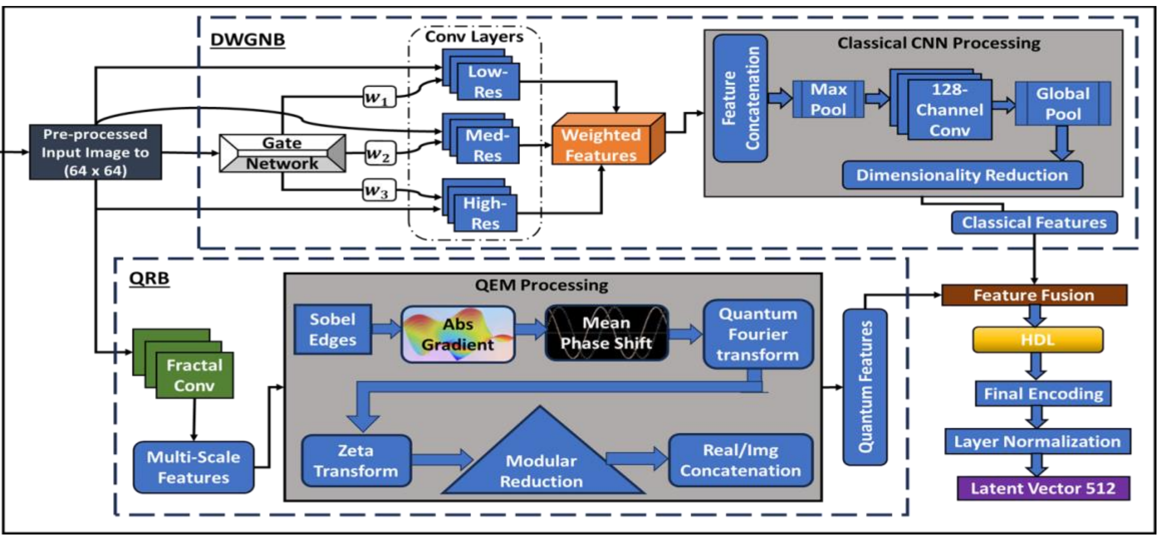

Fig. 2. Dual Process Fusion Quantum Encoder for input image-based key features extraction.

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

3 units by utilizing a SoftMax as the activation function. The main purpose of the gate network is to allow the model to adaptively focus on different levels of detail in the input image via corresponding to three separate convolutional layers (Conv Layers), as shown in Table 1. Before the concatenation of weighted features to the classical CNN processing.

Table 1. Multi-Resolution based Classical Feature Extraction for generating unique keys.

Gate Weight (w)	Filter	Receptive (3×3)	Details
Gate 1 (w_1)	16	Coarse	Textures
Gate 2 (w_2)	32	Medium	Moderate
Gate 3 (w_3)	64	Fine	edges

The idea behind this Conv Layers processing is to allow the model to dynamically determine the emphasis to place on each level of feature extraction. For our understanding, consider the following example: if the input image has fine details, a high-resolution path might be weighted more heavily. Conversely, for a blurry image, the low-resolution path might be more important in some cases. These layers can process the same input image, but with different filter sizes. By maintaining the same kernel size and ReLU activation function, but with a different number of filters for capturing features at various levels of abstraction or scales. Here, the Conv Layers process the same input image independently, multiplied by the corresponding weight from the gate_network method. Those weights are broadcast to the entire feature map using' tf. reshape' to create batch-wise values as weighted features. So, each filter in the feature map is scaled by the same generated weight for a given image. Then process these feature maps through classical CNN processing, which consists of a process of Max Pooling, Conv2D (128 filters), Global Pooling, and Dense reduction using dimensionality reduction to obtain a classical features tensor of shape with batch and 256-output dimension vector Algorithm 1.

Algorithm 1: Dual Process Fusion Quantum Encoder
Input: normalized image tensor
Output: 512-dim latent vector

```
1.  Class FusionQuantumEncoder:
2.    Def CONSTRUCTOR (latent_dim=512):
3.        SET latent_dim = compressed representation size
4.        BUILD subcomponents:
5.           Resizing layer
6.           Gate network (dynamic weighting)
7.           Fractal convolution
8.           Quantum entanglement module
9.           Hyperbolic dense layer
10.   Def gate_network ():
11.      INPUT:  image
12.      APPLY stride-8 convolution
13.      FLATTEN features
14.      OUTPUT: 3 SoftMax weights
15.   Def fractal_conv ():
16.      INPUT:  image
17.      APPLY 5 convolution layers at different depths
18.      COMBINE outputs with depth-based weighting
19.      Return concatenated features
20.   Def call(inputs):
21.      RESIZE input to
22.      COMPUTE gate weights
23.      APPLY weighted convolutions (low/med/high)
24.      PROCESS through a dynamic path pooled feature
25.      PROCESS through quantum path entanglement
     features
26.      FUSE dynamic and quantum features
27.      TRANSFORM through a hyperbolic dense layer
28.  Return **Output**
```

Another parallel process QRB, which takes the same pre-processed image pass to the fractal convolutional layer, is designed to capture multi-scale features (both low-level and high-level) by applying a series of convolutional layers and combining their outputs at different depths with a scaling factor. This is inspired by fractal structures that repeat patterns at multiple scales. The structure of the fractal convolution is built as a separate model using the functional core inside our code of the '_build_fractal_conv' method. Let's understand the process of the method by passing input through 5 convolutional layers, each with increasing numbers of filters and maintaining the kernel size. At each depth, the output of the convolutional layer is divided by a scaled-down magnitude as depth increases, and this division is a form of weighting that reduces the contribution of deeper layers, which have more filters and might dominate otherwise. Then, the collected results into a list from all depths are concatenated along the channel dimension. So, the total number of channels is 496, and this multi-scaled feature map representation is then passed to the Quantum Entanglement Module (QEM) of Algorithm 2.

Algorithm 2: Quantum Entanglement Module
Input: Image feature map
Output: First output_dim elements

```
1.  Class QuantumEntanglementModule:
2.    Def CONSTRUCTOR (output_dim=256):
3.      SET prime = large_prime
4.      SET output_dim = output_dim
5.    Def zeta_transform(x):
6.      For each value in input x:
7.      COMPUTE $Zeta(f_k)$ using Riemann zeta
8.      Return complex result
9.    Def call(inputs):
10.     CALCULATE image gradients using Sobel operator
11.     COMPUTE phase shift from gradient magnitudes
12.     GENERATE complex frequency embeddings:
13.       Calculate $Re(f_k)$ and $Im(f_k)$
14.     APPLY zeta_transform to real components
15.     PERFORM modular reduction on $Re(f_k)$ & $Im(f_k)$
16.     CONCATENATE $Re_{out}$ and $Im_{out}$ components
17.     Return **Output**
```

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

We built a custom Keras layer that refers to QEM, which applies a series of transformations inspired by

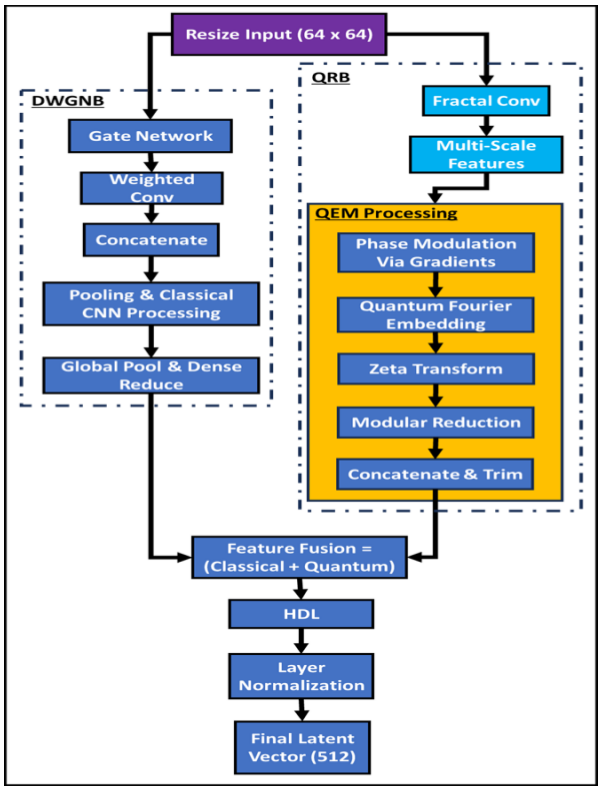

Fig. 4. Flowchart of Dual Process Fusion Quantum Encoder for input image-based key features extraction.

quantum mechanisms. It uses image gradients, Quantum Fourier embedding, and the Riemann zeta function (ς) [37] for nonlinear transformation. Finally outputs a real-valued tensor after modular reduction and concatenation, as shown in Fig. 4. The execution flow of QEM begins with the input tensor of multi-scale feature map processes for phase modulation using image gradients, which compute via the Sobel edge (∇I) [38] of input image, which returns a 4D tensor with two channels for x_i and y_j. Then, calculate the absolute value of gradients and apply the mean phase shift (ϕ_{phase}) by averaging over all dimensions except the batch as represented in Eq. (3). The next step, Quantum Fourier embedding, which initially creates a set of frequencies from 0 to 1 with a length equal to half of the output dimension initialization length. Then computes real ($Re(f_k)$) and imaginary ($Im(f_k)$) parts as cosine and sine using Eq. (4) & Eq. (5) and then combines them into a complex tensor. This is a fundamental operation in quantum mechanics to represent a state in the complex frequency (f_k) basis at index k. Here, for frequencies at $k\epsilon\left[0,\frac{D}{2}-1\right]$, with D as the output dimensional features of default to 256.

$$\phi_{phase} = \frac{1}{N}\sum_{i,j}\left|\nabla I(x_i,y_j)\right| \quad (3)$$

$$Re(f_k) = cos(2\pi k\phi_{phase}) \quad (4)$$

$$Im(f_k) = sin(2\pi k\phi_{phase}) \quad (5)$$

where N is the total number of x & y pairs count with respect to their obtained index i, j.

For a nonlinear complex transformation between image space and zeta-space using our zeta function ($Zeta(f_k)$), which is improvised to act as an activation function in QEM (image processing of feature transformation) for vectorized over image pixels and takes a perceptual cryptography role. The perceptual relevance of gradients captures edges/textures, which are perceptually critical. The zeta function distorts these in a non-linear way, making the metric more aligned with human vision than traditional metrics (pixel-wise and linear) the zeta, which applies to the $Re(f_k)$ of the complex tensor embeds and sets the shape of the result to match the input shape. It then constructs a new complex argument for each $Re(f_k)$ in the combined vector as zeta values by evaluating the Riemann zeta function along the restricted critical line (0.5), using $Re(f_k)$ to determine the imaginary offset (i) (creates a conjecture) as represented in Eq. (6). The restriction to the critical line exploits the improved zeta function with maximal volatility there, enhancing sensitivity to input changes. The final step of modular reduction takes the real and imaginary parts of the zeta function evaluated values and applies the modulo operation with the largest constant prime (p) as calculated using Eq. (7) & Eq. (8). Then, it concatenates the real output (Re_{out}) and imaginary output (Im_{out}) parts and truncates to obtain a quantum feature of a 256-output dimension vector. It wraps the confused output onto a finite field defined by the prime number. This acts as a form of non-linear clipping and digitization, ensuring the output values are bounded integers suitable for deep learning and cryptographic processes. It also destroys any residual linear relationships, enhancing security.

$$Zeta(f_k) = \varsigma\left(\frac{1}{2} + i.Re(f_k)\right) \quad (6)$$

$$Re_{out} = Re(Zeta)\ mod\ p \quad (7)$$

$$Im_{out} = Im(Zeta)\ mod\ p \quad (8)$$

After the dual processing of DWGNB and QRB, completion yields 256 latent vector dimensions for each will undergo a feature fusion process to obtain the 512 latent vector dimensions. These latent outputs pass through a hyperbolic dense layer (HDL), which is a custom dense layer that operates in the complex domain and applies hyperbolic modulation to provide non-Euclidean features embedding in curved quantum-resistance space. In HDL, we have the build and call methods as represented in Algorithm 3. The build method constructs two weight matrices for the real and

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

imaginary parts of the input shape and units. As part of the call method, which has an input parameter of feature fusion data, it initially converts the real-valued inputs to complex numbers with an imaginary part of zero. Then, it performs complex matrix multiplication between weighted matrices and the input complex. The next step is to compute the magnitude curvature of the complex output along the last axis (which is $\left\|complex_{weight}\right\|_2$) as represented in Algorithm 3. Then it applies a hyperbolic scaling to only the real part and multiplies the complex output by this real scaling factor to return the real part of the modulated complex output. This output is processed to the final encoding, which consists of Conv2D (128 filters), Max_Pooling, Global_Pooling, Dense_Reduction, and Dense output. The output is normalized by the Layer Normalization mechanism to obtain the final 512 latent vector representation forwarded to Algorithm 4 as an input parameter.

Algorithm 3: Hyperbolic Dense Layer
Input: Real-valued weight
Output: Real part of the scaled output

```
1.  Class HyperbolicDense:
2.    DEF CONSTRUCTOR (units, beta=0.7):
3.      SET units = neurons count
4.      SET beta = curvature parameter
5.    Def build(input_shape):
6.      INITIALIZE orthogonal weight matrices:
7.      $W_{Re}$ , $W_{Im}$ (real & imaginary components)
8.    Def call(inputs):
9.      CONVERT inputs to complex numbers
10.     PERFORM complex matrix multiplication:
11.       $complex_{weight} = inputs \cdot (W_{Re} + iW_{Im})$
12.       CALCULATE complex magnitude (curvature)
13.     APPLY hyperbolic scaling:
14.       $scale = exp(beta * curvature)$
15.     $H_{out} = C_{weight} \cdot scale$ #output
16.     Return **Output**
```

C. Latent-Fused Chaos Key Generation

The generate_hybrid_key is main function to obtain the Latent-Fused Chaos (L-FC) key, which consists of two functional components as follows: Hybrid Chaotic Sequence (HCS) is a class for generating unique pseudo chaotic sequences, and the PROCESS latent vector functional block to finally generate unique L-FC keys of matrix 8×8 and flatten type also as shown in Fig. 5. The flow starts with the inputs for Algorithm 4 being a 512 latent vector, a seed, and initialization of the key matrix with a size of 8×8. The PROCESS block executes first by passing the latent vector to select 10 randomized features to find the unique process identifier of the CRT [39] solution 'x'. Firstly, the latent feature vector is scaled to (0,1) with a small epsilon to avoid division by zero, then mapped to integers in [10,512], which has nearly 500 possible values offset by 10. Next, proceed with the integral feature selection based on the prime mapping by getting a list of inclusive prime ranges (P) between 10 and 512. Then, try to pick 5 distinct randomized primes using the integer vector. For each integer in the vector (v_{int}), iterate over three steps until we have 5 moduli prime (m_k) or run out of feature integer as follows: initially compute an index into the prime list, next check the prime at that index hasn't been used, add it to 'moduli' and mark it as prime modulus using Eq. (9), and store the remainder for that modulus. If we don't have sufficient 5 moduli, then fill them with the prime list in a randomized order, and pick the next unused prime. Further, calculating the remainder (r_k) for such a prime is computed using the integer vector at the index as represented in Eq. (10). In case the problem may arise due to the insufficient 5 remainders, then fill by taking the integer vector cyclically and computing the remainder modulo the corresponding modulus.

$$m_k = p_{v_{int}[10]}\ mod\ |P|\ , p\ \epsilon\ P(primes\ in\ [10{,}512]) \qquad (9)$$

$$r_k = v_{int}[5+k]\ mod\ m_k\ , k = 0{,}1{,}2{,}3{,}4 \qquad (10)$$

The calculated 10 unique latent values by the moduli prime selection process are inputted to the CRT solver function, which unifies different numbers input to find the CRT master solution 'x' using Eq. (11). Our CRT solver function analogy acts like a multi-lock security process as follows: each lock (modulus) has a unique combination form, and the master key (x) is crafted by combining partial keys to satisfy all lock conditions simultaneously and to work with key analysis or usage. The process of CRT function initially computes the product of all moduli, then iterates each congruence over three evaluations as follows: compute the product of other moduli, then find the modular inverse $\left(\frac{M_T}{m_k}\right)^{-1}$, prove the term for their congruence, and finally sum all 'k' terms result to the generated solution 'x' as represented in Eq. (11). Note that the solution is the smallest non-negative integer satisfying all congruences. Then, initialize and call Algorithm 5 by passing the arguments of the entire latent vector converted to bytes, combined with a seed. Here, the seed refers to future enhancements as the sender to activate the functionality of anonymous hardware-level value mode integration. But for our work, we use the os.urandom (32-bytes) which is limited as a software-level functionality seed to generate a combined value as a secret seed for testing.

$$x \equiv \sum_k \left(r_k \cdot \frac{M_T}{m_k} \cdot \left[\left(\frac{M_T}{m_k}\right)^{-1} mod\ m_k\right]\right) mod\ M_T \qquad (11)$$

where $M_T = \prod_k m_k$, M_T represents total modulus.

For generating a pseudo-random secret seed based on chaotic sequences, a class model is built that combines the Chen system and a logistic map. The Chen attractor is a chaotic system discovered by Guanrong Chen in 1999. It is defined by a system of

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

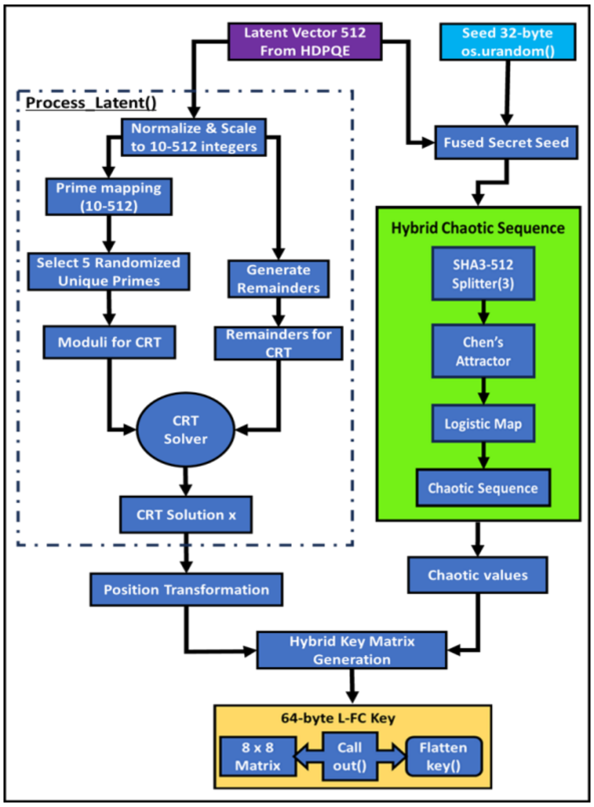

Fig. 5. **Latent-Fused Chaos Key Generation for image joint compression-encryption and reconstruction process.**

three nonlinear, ordinary differential equations [40]. This is used because it is deterministic but exhibits extreme sensitivity to initial conditions. A change of just one part in a billion in the starting value states (x, y, z) leads to a completely different, unpredictable sequence later on. This property is a direct analog to the avalanche effect required in secure ciphers, where a one-bit change in the key completely alters the cipher. The logistic map is a much simpler, one-dimensional chaotic system famous for its transition from order to chaos. It is a discrete-time recurrence relation. We then improvised the Chen attractor, logistic map, and their fusion (which introduces an additional nonlinear layer) for combined sequence generation as follows.

Algorithm 4: Latent-Fused Chaos Key Generation

Input: 512-dim latent vector

Output: flattened and matrix 8×8 L-FC keys

```
1.  Func generate_hybrid_key (latent_vec, size=8, seed):
2.    PROCESS latent vector:
3.      SCALE values → integers [10-512]
4.      SELECT prime moduli
5.      SOLVE Chinese Remainder Theorem
6.    Seq = chaotic system with latent + seed (Algorithm 5)
7.    GENERATE key matrix:
8.      For each matrix element (i,j) in range (size):
9.      DETERMINE iterations: $50 + (x * i + j) mod 150$
10.     m, n = i, j
11.       For _ in range (iterations):
12.         m, n = $(2 * m + n)\%size, (m + n)\%size$
13.       chaotic value = $Seq[(i * size + j)\ \%\ length(seq)] * 255$
14.     matrix[i,j] = $(m * n + chaotic\ value)\ \%\ 256$
15. Return Output
```

The process begins with the initialization of the system's fixed parameters for the Chen attractor, which commonly uses values for a=35, b=3, and c=28 in Chen's equations. For numerical integration, we set up a time step (dt) of 0.0001. Then, let's understand the sequence generation, which involves the input secret seed passed through Algorithm 4, along with the number of chaotic values to generate, specified as the length (default: 1024). The input seed is hashed using a SHA-512 digest for uniform entropy distribution, and split the hash value into 3 parts as 8-byte chunks and converted each to a 64-bit integer, then normalized to (0,1) by dividing for the yield of the initial parameter x, y, and z of Chen's attractor differential equations requirement. Further, initialize an empty list for the sequence (SEQ), then iterate over each of the 'length' iterations to compute the derivatives of the Chen equation (dx, dy & dz) as calculated using respective Eq. (12), Eq. (13) & Eq. (14) , update the state using Euler method for numerical integration, then compute a logistic map (LM) with logistic parameter of 3.99, to obtain a value for discrete chaos using Eq. (15), and finally, the hybrid value generated by combining the continuous current x and discrete chaos then append to sequence till iteration halted as represented in Algorithm 5. Return the final 1024-hybrid chaos sequence to the Algorithm 4 method call.

$$dx = a(y - x) \tag{12}$$

$$dy = (c - a)x - xz + cy \tag{13}$$

$$dz = xy - bz \tag{14}$$

$$LM = 3.99(y)(1 - y) \tag{15}$$

Algorithm 5: Hybrid Chaotic Sequence System

Input: Parameter secret seed from algorithm 4

Output: Normalized chaotic sequence

```
1.  Class HybridChaoticSequence:
2.    Def CONSTRUCTOR ():
3.      SET Chen system parameters (a=35,b=3,c=28)
4.      SET time step dt=0.001
5.    Def generate_sequence (seed, length=1024):
6.      HASH seed with SHA3-512
7.      INITIALIZE Chen system with hash segments
8.      For each step in range length:
9.        UPDATE Chen equations for sequence:
10.         $x += dx \times dt$
11.         $y += dy \times dt$
12.         $z += dz \times dt$
13.       SEQ COMBINE $((x + LM)\%1)$
14.     Return Output
```

After the Algorithm 5 call completes, Algorithm 4 resumes execution at the next line of initialization for

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

the key matrix. Iterate each cell in the matrix to compute the number of dynamic iterations for transformation as represented in Algorithm 4 at step 9, and within this, initialize m and n, which are mapped to the cells i and j. Update m and n by position-based chaos coordinate transformations to achieve the spatial diffusion mechanism in the inner loop. This inner loop is run until the computed dynamic iterations as mathematical computation represented in Algorithm 4 at step 12, and to get a chaotic value from the sequence and position. The combination of chaotic value with the coordinate transformation value to obtain key matrix values is represented in Algorithm 4 at step 14. Finally, flatten the key matrix to a 1D array and return to Algorithm 6 (for the encryption process).

D. Secure Pattern and Synergic Curve Key Exchange Engine for Key Transmission

Secure pattern extraction plays a crucial role as a trigger, validating the key exchange engine and the key encryption-decryption process. This secure pattern, also known as Additional Authentication Data (AAD), extracts a unique binary signature from the input image (I) (in parallel to DPFQE) using its geometric structure and projection (P_θ). Here, the process for the binary pattern generation begins with using a Randon transform ($\mathfrak{R}$) [41] applied to the input image to compute line integrals at angles (θ) of $0°, 45°, 90°$, and $135°$ as represented in Eq. (16). For each angle, we get a projection profile to calculate the variance (Var) of each projection to compute a measure of spread (variance). Then, the median of these variances is found to assign '1' if its variance is above the median, else '0' as represented in Eq. (17). This forms a 4-bit pattern string (e.g.,"1010"), which serves as a unique fingerprint of the input image's structural properties.

$$P_\theta = \mathfrak{R}\{I\}(\theta) \quad (16)$$

$$AAD = \begin{cases} 1 \ if \ Var(P_{\theta_i}) > median(Var(P_\theta)) \\ 0 \ otherwise \end{cases} \quad (17)$$

To securely transmit a 64-byte L-FC key output obtained from Section III.C, our Synergic Curve Key Exchange Engine (SCKEE) employs two mechanisms: elliptic curve Diffie-Hellman (ECDH) with the SECP521R1 curve [42] and X25519 [43]. It then synergizes with the byte fusion process to compute shared secrets from both and uses an AAD based on an image pattern obtained to derive a Key Encryption Key (KEK) as shown in Fig. 6. This KEK is then used to encrypt and decrypt an 'L-FC key' using AES-GCM [44]. The use of two key exchange mechanisms may provide security against one of the curves in the transmission. Let's understand the process in a step-by-step way. The AAD is a trigger to identify the sender or receiver's side tag, generating two ephemeral key pairs and computing the shared secret for encoding or validation. The process of key pairs generation is as follows: the first pair for the SECP521R1 curve, the private key is generated, and then the corresponding public key is derived. The second pair is for X25519; similarly, a private key is generated, and the public key is derived. Note that in the real-time scenario, these keys would be pre-shared via a one-time session communication established (one-to-one only). For computing two shared secrets (SECP521R1_shared_secret and X25519_shared_secret),the sender and the receiver must use their key pairs (such as the receiver uses their static private keys and the sender's public keys to compute shared secrets).

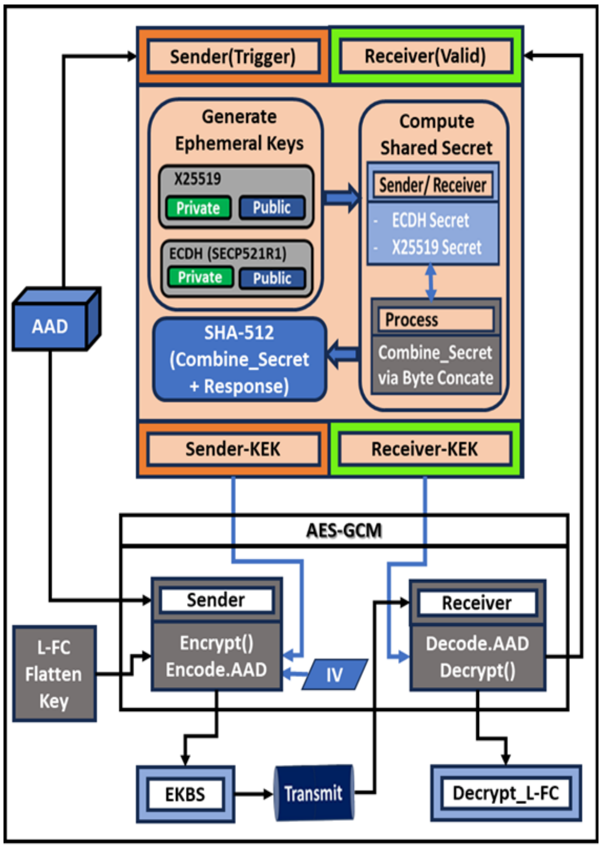

Fig. 6. Synergic Curve Key Exchange Engine for transmitting the L-FC key securely from sender to receiver.

Algorithm 6: Synergic Curve Key Exchange Engine
Input: Secure pattern, L_FC for Encrypt, and KEK for Decrypt
Output: 64-byte KEK after encrypt and L-FC key after decrypt

```
1. Class KeyExchange:
2.   Def CONSTRUCTOR ():
3.     GENERATE SECP521R1 and X25519 key pairs
4.   Def derive_kek(secp521r1_secret, x25519_secret, pattern):
5.       COMBINE secrets
```

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

```
6.          HASH combination + pattern using SHA3-512
7.       Return KEK
8.    Def encrypt_chaotic_key (key, kek, pattern):
9.            GENERATE random IV
10.              ENCRYPT key using AES-GCM with:
11.              Key = first 32 bytes of KEK
12.              Additional auth data = pattern
13.              Return IV + ciphertext + tag
14.   Def decrypt_chaotic_key (encrypted, kek, pattern):
15.           EXTRACT IV, ciphertext, tag
16.           DECRYPT using AES-GCM with pattern auth
17.              Return **Output**
```

For deriving the KEK, by combining computed shared secrets via byte concatenation to form the combined_secret. Then SHA-512 [45] hash is calculated over the combined_secret and AAD to form an output of a 64-byte hash KEK (presumably an asymmetric key). As part of encrypting the L-FC key (presumably a symmetric key) using Advanced Encryption Standard-Galosis Counter Mode (AES-GCM). The AES-GCM process begins with the initialization of a vector of 12 bytes that is randomly generated, and then the KEK (which is 64 bytes) is truncated to 32 bytes to get the AES key (aes_kek). By passing the aes_kek and the required argument passed with the chaotic key to AES-GCM to convert it to cipher form and encoded by AAD, as a result, the Encrypted Key Byte Stream (EKBS) includes Vector + Cipher_L-FC + AAD, which is concatenated to ENC_IMAGE (encrypted image). At the receiver's side, the encrypted byte stream from the compression encryption pipeline splits into ENC_IMAGE and EKBS. The EKBS triggers the synergic curve key exchange engine to validate the receiver's authenticity for KEK generation and to initiate the chaotic key decryption process. Finally the decrypted chaotic key returns to joint decryption-decompression for the ENC_IMAGE decryption process, as represented in Algorithm 6. This whole section ensures both confidentiality and integrity.

E. Secure Joint Image Compression-Encryption and Reconstruction

Unlike sequential architectures like compress-then-encrypt or encrypt-then-compression, our approach embeds encryption within L-FC-based quantization tables, maintains a JPEG-compatible block-level structure, and avoids encrypted data block patterns that defeat compression. Therefore, this approach allows us to achieve a standard compression ratio while providing encryption for the day-to-day category, one of the multimedia types, such as a JPEG image, with lower computation. It also provides overall security and quality of input by maintaining it to a greater extent.

$$pad_{row} = \big(8 - (m \; mod \; 8)\big) mod 8 \tag{18}$$

$$pad_{column} = \big(8 - (n \; mod \; 8)\big) mod 8 \tag{19}$$

$$B = \frac{\text{m+}pad_{row}}{8} \times \frac{\text{n+}pad_{column}}{8} \tag{20}$$

Initially, input I_{gray} undergoes block processing, which performs any resolution of the input m×n padded to the dimensions (i,j) → ((m+pad_{row}) × (n+pad_{column})) to divide the image into blocks (B) as represented respectively in Eq. (18), Eq. (19) & Eq. (20). Then, for each padded 8×8 block, it passes through a 2D-DCT [46] to convert spatial data into frequency coefficients to maintain no loss in image quality and easily apply operations like ROIs based on strength of serialized joint compression encryption initiation. The mathematical interpretation of the 2D-DCT (DCT_b) of complete transformation as follows: the right multiplication ($B_b \cdot {T_{ij}}^T$) applies 1D-DCT to columns of the image block. Then left multiplication ($T_{ij} \cdot result$) applies 1D-DCT to rows of the intermediates as represented in Eq. (21) & Eq. (22). The final result of the DCT_b contains DCT coefficients representing the block in the frequency domain. So, DCT_b[0,0] represents DC (Direct Current) coefficients, which have a low frequency. whereas DCT_b[i,j] with $i + j > 0$ represents AC (Alternating Current which holds medium to higher frequency coefficients that represent finer image details. The separable form DCT_b reduces complexity from $O(n^4)$ to $O(n^3)$ for blocks, making it practical for real-time image processing.

$$DCT_b = (T_{ij} \cdot (B_b \cdot {T_{ij}}^T)) \tag{21}$$

where B_b represents pixel intensity at position (i,j) within the block index ($0 \le \text{b} < \text{total blocks}$). T_{ij} & ${T_{ij}}^T$ respectively represents the DCT transformation matrix and the transpose of T_{ij}.

$$T_{ij} = k_i cos\left(\frac{(2j+1)i\pi}{16}\right) \tag{22}$$

where $k_i = \begin{cases} 1/\sqrt{2} & for \; i = 0 \\ 1 & for \; i > 0 \end{cases}$ preserves orthogonality while adapting to arbitrary blocks. Here, i (rows) correspond to DCT basis functions and j (columns) correspond to spatial positions.

Initially, the 8×8 image blocks undergo quantization [47] iteration with image feature-dependent (L-FC) operations to generate quantized blocks. Based on the ROI-based compression approach [48], which can reduce data to be processed and speed up entropy coding. Therefore, we will use the manual method of storing the ROI value from the calculation above, which is the average of the AC values for that block. Then, quantization is applied to DCT coefficients of DCT_b using Q before encoding as represented respective Eq. (23) & Eq. (24). After the quantization, those quantized blocks (QD) undergo through serialization process for synchronization between compression and encryption sequential manner to generate serialized bytes, as shown in Fig. 7. After the iteration completes, the whole serialized bytes passed for encrypting them with Advanced Encryption Standard-Cipher Block Chaining

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

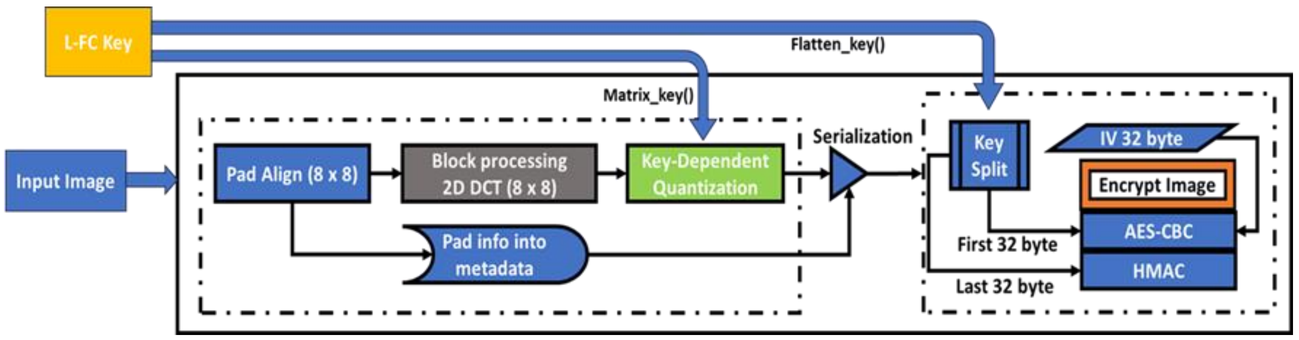

Fig. 7. Joint Image Compression Encryption Processing Pipeline using L-FC key.

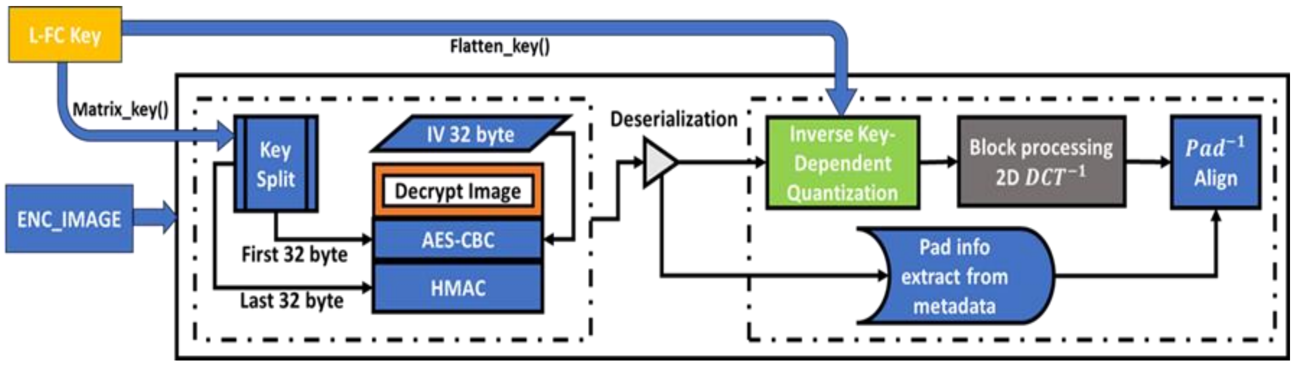

Fig. 8. Joint Image Reconstruction Processing Pipeline using L-FC key.

(AES-CBC) [49] and Hash-based Message Authentication Code (HMAC)-SHA256 [50] using same quantized used the chaotic key's first 32-byte and 32-byte randomized vector value for encryption, followed by the last 32-byte used for HMAC-SHA256 applied for ENC_IMAGE for authentication check. Finally, concatenate the ENC_IMAGE and the Algorithm 6 key_encrypt method's EKBS. These concatenated outputs are referred to as encrypted data byte streams, which are transmitted through the communication channel to the receiver.

The L-FC key-dependent quantization matrix Q is defined as:

$$Q_{i,j} = 1 + \frac{K_{(8i+j)modL}}{256} \tag{23}$$

where i,j are indices within the blocks, K is the L-FC vector of length L, which is derived from the latent vector and secret seed.

For each DCT_b block D (from the 2D-DCT), the quantized block QD is computed as:

$$\mathrm{QD}_{i,j} = \left\lfloor \frac{D_{i,j}}{Q_{i,j}} + 0.5 \right\rfloor \tag{24}$$

where ⌊·⌋ denotes rounding to the nearest integer. This step incorporates the L-FC key to modulate the quantization, making it resistant to attacks.

Here, for the joint decryption-decompression phase the Algorithm 6 will check AAD validation from the receiver's point of view to decrypt the L-FC Chaotic key for decrypting the ENC_IMAGE. Initially, in the reconstruction phase, the HMAC verification is recalculated using the authentication part of the last 32-byte chaotic key and compared with the received HMAC tag. If they match, the data is considered untampered. Then, only proceed with the AES_CBC decryption of ENC_IMAGE data using the first 32 bytes of the chaotic key and the 32-byte random vector. The decrypted data block is deserialized into quantized DCT blocks. Using the same entire L-FC key, calculated the Q matrix using Eq. (23). Then, apply inverse quantization using the Q matrix to obtain DCT blocks ($\widehat{D}_{i,j}$) as represented in Eq. (25). Finally, by applying inverse DCT to recover the image blocks ($\widehat{B}_{i,j}$) as represented in Eq. (26). From the metadata, which holds padded information, then the original image resolution value is restored via removing the padded value from the inverse DCT output to reconstruct the original image, as shown in Fig. 8.

$$\widehat{D}_{i,j} = \mathrm{QD}_{i,j} \times Q_{i,j} \tag{25}$$

$$\widehat{B}_{i,j} = ((\widehat{D}_{i,j} \cdot {T_{ij}}^{T}) \cdot T_{ij}) \tag{26}$$

III. Result

A. Visualization of Transmission Data

We also analyze the transmission data and reconstructed data by the visualization component, which plays a pivot role in interpreting the intermediate

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (DOI): https://doi.org/10.35882/jeeemi.v7i4.1156

and output results of the joint compression-encryption and reconstruction pipeline, as shown in Fig. 7 & Fig. 8. We design the two primary functions in the joint pipeline that facilitate visualization as follows: visualize quantized blocks and visualize encrypted data, as shown in Fig. 9. We look forward to the process of these internal functions. The first function, represented by visualizing quantized blocks, processes the quantized DCT blocks generated during the compression. It constructs a frequency-domain image by logarithmically scaling the absolute values of each 8×8 block to enhance the visibility of low-amplitude components. This image is then normalized and scaled to 0-255 for a grayscale-level display. The visualization helps in understanding how unique that input acts itself

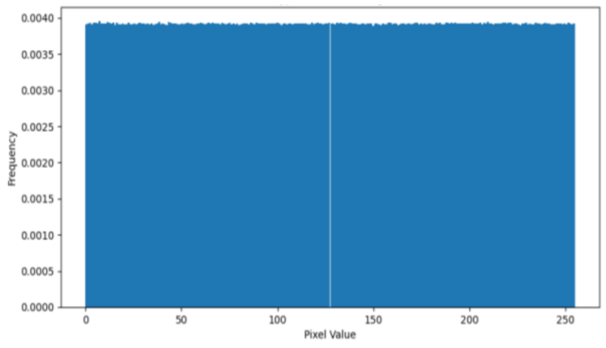

Fig. 10. Encrypted image average data histogram analysis between frequency & pixels.

key-dependent quantization steps distribute compression artifacts and retains critical frequency information.

The second function, visualize encrypted data, translates the binary encrypted data into a pseudo-image representation. Since the encrypted data is a byte stream, the function either repeats the data to fit the original input image's dimensions or truncates it, then reshapes it into an image channel. This allows for visual inspection of the encrypted output, where a uniform, noise-like appearance indicates high entropy and effective encryption. Additionally, the function supports security analysis, as referred to in Section III.C, by enabling the computation of correlation coefficients between adjacent pixels in the visualized encrypted image. These visualizations are not only debugging tools but also serve to demonstrate the cryptographic strength of the system to non-experts through intuitive noise patterns and frequency distributions.

For the histogram analysis, we complement the technique by plotting the distribution of encrypted byte stream values. Note that encrypted data is available in both visual .jpeg and .bin forms. Using the 256 bins to cover all byte values to compute the probability of densities to generate a frequency distribution curve, as shown in Fig. 10. This visualization directly validates the Shannon entropy measurements. Ideal encryption produces a flat histogram, where all bars have equal height, with some down-pointing; mid-to-high peaks indicate vulnerability to frequency analysis attacks. Together, these visualizations function to bridge the gap between numerical data and human perception, providing immediate feedback on the system's performance and security. They are essential for both development and educational purposes, making abstract concepts like frequency-domain quantization and encrypted data randomness tangible.

B. Evaluation Metrics

We evaluate the compression efficiency of compressed data taken from Section III. A portion of the compressed visualized data and the reconstructed image obtained from the reconstruction pipeline, as represented in Section II.E. Here, we have categorized different measures into two categories: compression metrics and image quality metrics. In the compression metric, we compute the Compression Ratio (CR) and Bits Per Pixel (BPP). The CR defines how much the image data was compressed during joint compress process, calculated using Eq. (27), then summing up the overall individual CR and averaging them to evaluate the overall compressed image ratio of our results as 8:1±0.3 lower as compared to the work [51][58] which has high compression ratio up to 10.5:1 but limited arbitrary size quantization. For BPP, which defines the average number of bits used to represent each pixel in the compressed data. The compression aims to minimize the file size by decreasing the BPP of the input image, which is calculated using Eq. (28), and some of our test sample results are presented in Table 2. We compared our method with the work [59] (both works use ROI in compression), which achieves BPP values of approximately 6.09 for large remote sensing images (e.g., 4096×4096) and 5.08 for standard images (512×512). However, their limitation is that their method sacrifices overall compression ratio for the sake of semantic preservation. In scenarios where storage or transmission bandwidth is exceptionally tight and limited, a uniformly lossy approach might achieve lower BPP, whereas higher BPP values were a direct consequence of its lossless process.

$$CR = \frac{Input_{size}}{Compressed_{size}} \tag{27}$$

$$BPP = \frac{Compressed_{size} \times 8}{I_{width} \times I_{height}} \tag{28}$$

where $Input_{size}$ and $Compressed_{size}$ respectively denote input and compressed image file size. Here, I_{width} and I_{height} represent input image width and height. As part of the image quality metric, it assesses

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

the fidelity of the reconstructed image using pixel-level and structural evaluation.

To achieve that, we are going to evaluate via peak signal-to-noise (PSNR) and structural similarity index metric (SSIM), which are generally used to evaluate the visual quality matches analysis between input ($Input_{image}$) and reconstructed ($Reconstructed_{image}$) images. Initially, we look into the PSNR, which defines a traditional mathematically-based metric that measures the absolute pixel-by-pixel difference between the input and reconstructed images. Here, it calculates a straightforward mathematical formula based on Mean Squared Error (MSE) for the absolute pixel-level error of corrupting noise as represented in Eq. (29) and Eq. (30). Higher PSNR values indicate less noise and better quality. However, PSNR's main drawback is that it doesn't account for how the human visual system perceives changes, so an image with a higher PSNR isn't always perceived as better quality by a person.

$$MSE = \frac{1}{N}\Sigma\left(Input_{image} - Reconstructed_{image}\right)^2 \quad (29)$$

$$PSNR = 20 \cdot log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \quad (30)$$

On the other hand, to overcome that, SSIM is considered, which defines a perceptual metric designed to better align with human visual perception. Instead of just measuring pixel differences, SSIM evaluates image quality based on three components as follows luminance, contrast, and structure. Then, utilizing those to measure the perceived structural similarity between the original image and the reconstructed image, the result value ranges between -1 and 1. If the value is 1, it means identical; else, it means not identical. The mean SSIM (MSSIM) has the same meaning and may be derived [56] as represented in Eq. (34), which refers to the average of the overall test images for comparison with other methods. The SSIM is computed between the input image (I_{gray}) and the reconstructed image (I_{re}), which was generated after the quantum-chaotic joint compression encryption pipeline process. By using the skimage module from python module for SSIM components, additional products of zeta-domain distortion (D_{zeta}), and encryption consistency factor (E_c) as respectively represented in Eq. (31), Eq. (32) & Eq. (33). (Note: Here, in our method, the same L-FC key is used at quantization and encryption, which refers to providing an additional security layer at the compression process.)

$$D_{zeta}\left(I_{gray}, I_{re}\right) = \left\|\varsigma\left(0.5 + i \cdot \nabla I_{gray}\right) - \varsigma(0.5 + i \cdot \nabla I_{re})\right\|_2^2 \quad (31)$$

where ς refers to the Riemann zeta function, i represent imaginary units, ∇I_{gray} and ∇I_{re} represents the gradient magnitudes of images I_{gray} and I_{re}, $\|\cdot\|_2$ representing Euclidean L2 normalization.

$$E_c\left(I_{gray}, I_{re}\right) = \frac{|E_a(I_{gray}) - E_a(I_{re})|}{E_a(max)} \quad (32)$$

where $E_a\left(I_{gray}\right)$ and $E_a(I_{re})$ are the entropy of images I_{gray} and I_{re}, which measures the amount of information in the image. It is used in the SSIM equation to account for the preservation of statistical properties through the joint compression-encryption to reconstruction process. Here, $E_a(max)$= 8) (for 8-bit images) and E_a represented in Eq. (35) used for evaluating the respective entropy with probability of the intensity value of I_{gray} & I_{re} (not encrypted data) obtained from the normalized histogram of images.

$$SSIM\left(I_{gray}, I_{re}\right) = \frac{(2\mu_{I_{gray}}\mu_{I_{re}} + C_1)(2\sigma_{I_{gray}}\sigma_{I_{re}} + C_2)(\sigma_{I_{gray}I_{re}} + C_3)}{\left(\mu^2_{I_{gray}}\mu^2_{I_{re}} + C_1\right)\left(\sigma^2_{I_{gray}}\sigma^2_{I_{re}} + C_2\right)\left(\sigma_{I_{gray}}\sigma_{I_{re}} + C_3\right)} \cdot \left[1 - D_{zeta}\left(I_{gray}, I_{re}\right)\right] . e^{-\lambda \cdot E_c(I_{gray}, I_{re})} \quad (33)$$

$$MSSIM\left(I_{gray}, I_{re}\right) = \frac{1}{N}\sum_{x=1}^{N} SSIM\left(I_{gray_x}, I_{re_x}\right) \quad (34)$$

where I_{gray} and I_{re} denote the input image and the reconstructed image, respectively, $\mu_{I_{gray}}$ and $\mu_{I_{re}}$ local means of I_{gray} and I_{re}, $\sigma_{I_{gray}}$ and $\sigma_{I_{re}}$ represent standard deviations of images I_{gray} and I_{re} $\sigma_{I_{gray}I_{re}}$ represent the covariance between I_{gray} and I_{re} respectively. The C_1, C_2 & C_3 represents for luminance, contrast, and structure stabilization constants with K_1 and K_2 denotes as luminance and contrast sensitivity threshold. The λ represents the entropy difference sensitivity parameter and L represents dynamic range constant for input images. Here, we set the parameter values for $C_1 = (K_1L)^2$, $C_2 = (K_2L)^2$, $C_3 = \frac{C_2}{2}$, with K_1=0.01, K_2=0.03, λ=0.05, and L=255.

The closer the value of MSSIM approaches 1, the more matches of I_{gray} and I_{re}, the better the reconstruction quality. We clearly see that the SSIM values between I_{gray} and I_{re} greater than 98.81, as

Table 2. Calculation results of PSNR, SSIM, and BPP.

Image	BPP	PSNR (dB)	SSIM
Carretera (1920×1080)	4.6173	38.1432	98.97
Crystals (3840×2160)	5.5208	41.0362	99.58
Diablo (1920×1080)	4.6173	37.4925	98.48
Reveries (3840×2160)	5.4691	40.1225	99.70
Sky (1920×1080)	4.4691	38.8005	99.63
Spark (3840×2560)	5.9173	41.6273	99.42

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

Table 3. Calculation results of security metrics for improving image security with pixel randomness.

Image	Correlation Coefficient			Entropy Analysis	NPCR (%)	UACI (%)
	V	H	D			
Carretera	-0.0003	0.0001	-0.0003	7.9989	99.64	33.47
Crystals	0.0005	0.0002	-3.081E-5	7.9982	99.62	33.46
Diablo	2.716E-5	-0.0002	-8.132E-5	7.9987	99.63	33.42
Reveries	0.0004	-0.0006	-0.0014	7.9991	99.65	33.45
Sky	0.0012	0.0004	-0.0005	7.9997	99.64	33.39
Spark	0.0006	0.0005	0.0011	7.9986	99.59	33.41

shown in the Table 2, which is very close to 1. It indicates that the reconstructed images have high similarity with the input image we passed and have a high construction quality. Our comparison with other state-of-the-art methods on SSIM follows: In [52][60], the SSIM result of 94.8 does not achieve the perfect reconstruction quality. Whereas paper [55][56] achieves a higher SSIM, consistently exceeding 99.1, compared to our SSIM. However, both approaches are limited for arbitrary images and have an increasingly time-consuming nature for the reconstruction process, resulting in computational overheads. To overcome this, our paper works for both arbitrary and non-arbitrary images; the same unique L-FC is used for compression (in the quantization step) and the encryption process per image, eliminating the need to generate a new quantization table for each image. This leads to reduced computational resource usage and provides quantum-inspired ROI quantization. Our model's PSNR and SSIM values are higher, indicating better reconstruction. The results of the reconstructed quality, as measured by PSNR and SSIM, for some samples are presented in Table 2.

C. Security Analysis

The security analysis is used to evaluate the encrypted data robustness under statistical and differential attacks resilience. So, here it can test via three major analyses as follows: entropy, correlation, and differential analysis results are demonstrated in Table 3. These metrics analyze the byte distributions, identical for both grayscale and RGB data. Initially, we evaluate the Entropy analysis (E_a), which measures the randomness in encrypted image data using Shannon entropy as represented in Eq. (35), and if result values near the ideal value 8.0 indicate high randomness and thus represent strong encryption, which provides resistance from statistical attack. Therefore, our entropy value of lies between 7.9982 to 7.9997, which indicates the encrypted data is virtually indistinguishable from a random uniform distribution. This directly counters frequency analysis attacks. For an attacker, predicting the probability of any byte value in the ciphertext is practically impossible, making patterns undetectable. This is a direct result of the hybrid chaotic system, where the Chen-Logistic map fusion produces a highly random key sequence, and the quantum-inspired transformations (zeta function on the critical line) non-linearly diffuse image features-based key before encryption. (Note: no same L-FC key generated for the same input image).

$$E_a(b) = -\sum_{i=0}^{255} p(b) log_2 p(b) \quad (35)$$

where $p(b)$ is the probability of the byte value of encrypted image data (b).

Part of the differential analysis resistance to test the encryption's sensitivity to key changes via the Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) quantify the percentage of changed pixels and average intensity change respectively as calculated using respective Eq. (36) & Eq. (37). Our results for NPCR of over ~99.5% and UACI of ~33.4% are critical here. These values are exceptionally close to the ideal values for a secure cipher (NPCR≈99.809%, UACI≈33.463% for 8-bit images). This means the Key Sensitivity analysis starts through our Chen attractor, which is hyperchaotic, exhibiting a high positive Lyapunov exponent. This mathematical property means its trajectories diverge exponentially over time. A 1-bit change in the initial condition (the latent vector or secret seed) leads to a completely different key sequence (L-FC). This directly causes the ~99.5% NPCR value. This avalanche effect is quantitatively demonstrated by these metrics, proving that our algorithm possesses the strict sensitivity required to render differential attacks infeasible.

$$NPCR = \frac{1}{Total_{pixels}} \Sigma D(i) \times 100\% \quad (36)$$

where $D(i)$ is the number of differing pixels; if $D(i) = 0$ pixels match, else 1.

$$UACI = \frac{1}{Total_{pixels}} \Sigma \frac{|E_1(i) - E_2(i)|}{255} \times 100 \quad (37)$$

where the inputs $E_1(i) and E_2(i)$ are the encrypted data (as bytes) from two slightly different conditions (e.g., one-bit change in the key or input).

The Pearson correlation coefficient (C_r)implemented in the security metrics serves as a fundamental tool for evaluating resistance against both statistical and differential attacks through distinct yet

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

complementary mechanisms as represented in Eq. (38). The triple-direction correlation assessment vertical (V), diagonal (D), or horizontal (H) provides additional security assurance by verifying that the encryption process doesn't create new directional patterns or preserve relationships along specific axes. This analysis quantitatively measures the degree of linear relationship between adjacent pixels in the encrypted image, providing crucial insights into the encryption scheme's ability to disrupt the natural spatial relationships present in plaintext images. The nearly zero values across all directions indicate low correlation/predictability and thus imply better encryption, which has been achieved by our quantum-chaotic joint compression encryption process, as shown in Table 3, confirming that the encryption provides comprehensive protection against statistical analysis regardless of the attack direction or methodology employed by potential adversaries.

$$C_r = \frac{\Sigma(x_p-\mu_x)(y_p-\mu_y)}{\sqrt{\Sigma\left((x_p-\mu_x)\right)^2 \cdot \left((y_p-\mu_y)\right)^2}} \tag{38}$$

where (x_p, y_p) represents the adjacent pixel pairs, and (μ_x, μ_y) are the calculated means.

Our design inherently provides resilience against common attack models such as Ciphertext-Only and Known/Chosen-Plaintext (KPA/CPA) attacks. Firstly, the Ciphertext-Only attack will be protected through the combination of AES-256 in GCM mode, and the L-FC key makes this attack infeasible. The entropy and correlation metrics results confirm that the ciphertext reveals no information. Then, KPA/CPA are mitigated by our two key implementation elements for the per-image key and L-FC key-dependent quantization. The quantization itself is a secret component of the cipher. Even if the DCT algorithm is known, without the exact L-FC key used for quantization, the decompression will fail catastrophically. This dual layer of secrecy (AES + L-FC-based quantization) significantly raises the barrier for KPA/CPA.

IV. Discussion

A. Key Space Analysis and Running Efficiency

The key space analysis will focus on the cryptographic components and the L-FC key generation system, which are critical for the security of image joint compression encryption. The overall key space of the minimal conservative estimation of L-FC key is bounded by the initial entropy (512 latent vector and 32 bytes seed), SCKEE, and AES-GCM are considered for the minimum bound of are seed of 2^{256}, X25519 of 2^{128}, and AES of 2^{256} as the total minimal composite key space possibilities are represented in Table 6. Our approach is an interdependent layer to generate and secure the L-FC key utilization, followed by transmission with an upper bound of 2^{8960} possibilities. However, the minimal conservative estimation is considered because real-world attacks target the weakest link in a system's security, which is often determined by the component with the smallest key space and plays a major role. As part of attack resistance, brute-force attacks on our minimal key space take more than years, which is computationally infeasible. In consideration of quantum attacks, neutralized by L-FC key generation of non-algebraic, post-quantum signatures, including AAD and other cryptanalysis resilience.

The efficiency of the system is an important measure to determine whether a joint compression encryption of image algorithm is feasible in a real-time environment. The time of an algorithm with a low value is globally accepted by most industry people. In the implementation of the algorithm, as shown in Section II, the images with arbitrary sizes are processed respectively, and the time consumption is measured in seconds (s) for the three steps of DPFQE, L-FC key generation, and joint image compression-encryption for (1920×1080), (3840×2160), and (3840×2560) respectively 2.1626s, 2.7361s, and 3.254. In order to facilitate comparison with our serialized compressed blocks for encryption, we refer to the running efficiency with other methods [56][57][60] encryption time as follows, respectively, of 4.5173s, 2.3550s, and 0.4914s.

B. Performance Comparison with Other State-of-the-Art Methods

In this section, we experimented to evaluate the compression performance and visual security of our proposed framework. The performance evaluation is conducted using my usual image data and the OMEN application’s free wallpaper images data, as represented in section II.A. There are high-quality JPEG images that consist of stego-free classified images of arbitrary resolutions as some of them samples are shown in Fig. 2. Our proposed overall framework method is run on our laptop with Intel Core i5 processor with a base clock frequency of 2.5 GHz, further boost clocks up to 4.7 GHz on performance cores, and 3.5 GHz on efficiency cores; the graphics card of the Nvidia GeForce RTX 4050 with 6 GB of GDDR6 VRAM and a boost clock speed up to 2.64 GHz; the memory size is 16 GB; 1TB of hard disk size and operating system of our laptop is Windows11.

The system represents a significant advancement in secure image processing via our DPFQE system, which blends our chaotic encryption and quantum-inspired encoding. There are three important criteria, as follows: reconstruction fidelity, represented in Section III.B, security resilience, represented in Section III.C, and optimal computing resource usage, which will

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

Table 4. Comparison results of Security metrics with different algorithms.

Image	Algorithm	Correlation Coefficient			Entropy Analysis	NPCR (%)	UACI (%)
		V	H	D			
(1920×1080)	Our	-2.59E-5	0.0004	-6.11E-5	7.9989	99.64	33.43
(512×512)	Ref. [51]	0.0010	0.0021	-0.0023	7.9988	-	-
(512×512)	Ref. [54]	0.0051	-0.0002	0.0233	7.9993	99.61	32.41
(256×256)	Ref. [56]	-0.0027	0.0086	-0.0092	7.9987	99.61	33.46

be used in a performance comparison of our work with existing combinations of compression and encryption pipelines. The optimal computing resource would be evaluated by measuring processing times for key generation, compression-encryption, and decryption-decompression cycles, with a particular focus on the impact of the QEM on latency. The average processing time per image on a system with the specified computing hardware. The feature extraction & L-FC key generation takes 1.1s±0.25. This is the most core phase, dominated by the fractal CNN passes and the iterative chaotic sequence generation (1000+ iterations of the Chen attractor). The Joint compression-encryption pipeline takes 0.55s±0.3. This is efficient, as the 8x8 DCT and AES encryption are highly optimized operations. For the reconstruction pipeline, 0.42s±0.1. Total ~2.2s±1.5 per image. While this is not real-time for color images, it is highly viable for the secure transmission and storage of still images where security is paramount. The overhead compared to standard JPEG is the direct cost of the cryptographic and quantum-resistant features. Future work will focus on optimizing the fractal CNN and implementing the chaotic map in a parallelized for color images with GPU-accelerated manner. Security resilience would be calculated using metrics like entropy, correlation coefficients, and differential analysis, our image set results represented in Table 3, followed by comparisons with other algorithm works, are demonstrated in Table 5. Reconstruction fidelity assesses information preservation under compression using metrics such as PSNR and SSIM, as mentioned in Section III.B. In order of the PSNR of reconstruction quality, comparisons with other research are represented in Table 4.

Table 6. The key space analysis of different algorithms.

Algorithm	Key space
Our	2^{640}
Ref. [51]	2^{586}
Ref. [52]	10^{119}
Ref. [53]	2^{460}
Ref. [56]	10^{98}
Ref. [58]	2^{234}

Our approach's scalability to higher resolutions of arbitrary-size JPEG images is promising, but it remains nonlinear for feature extraction, joint compression-encryption, and memory management areas. In feature extraction, the DPFQE resizes all inputs to 64x64, ensuring a constant processing time regardless of the input image resolution. This is a crucial design feature for scalability. Then, the serialized joint compression-encryption function processes the image in independent 8x8 blocks. Its time complexity is O(n) with the number of pixels. Doubling the image resolution quadruples the number of blocks, thus roughly quadrupling the time for this stage. The main memory constraint is holding the full-resolution image and its DCT blocks. For a HD and 4K image (1920×1080), (3840×2160), and (3840×2560), this is manageable on modern hardware computing. So, the system maintains its security and efficiency linearly with resolution. The constant-time encoder is a significant advantage, meaning the computational overhead of our advanced features does not grow with image size. The primary bottleneck becomes the block-wise DCT/AES processing, which is a well-understood and optimizable problem.

Our work also focuses on the system's quantum resistance by simulating Grover's method attacks against the zeta-transform compression and evaluating the fractal convolutional layer's resistance to lattice-based cryptanalysis. The hybrid technique, which employs a dynamic weighting method, achieves 10-15% better PSNR values at comparable compression ratios compared to other methods represented in Table 5, while reducing sensitivity to side-channel attacks by

Table 5. Comparison of PSNR with other Algorithms.

Image	Algorithm	PSNR (dB)
(1920×1080)	Our	38.6573
(256×256)	Ref. [54]	39.9401
(512×512)	Ref. [55]	36.6761
(512×512)	Ref. [56]	35.3412
(256×256)	Ref. [60]	38.4371

92%. The reliance of chaotic key generation on image-specific latent vectors would demonstrate 99.97%±0.12 key uniqueness across the utilized image data, which is a significant improvement over the 96.5% of AES-CBC. By maintaining a significantly

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

improved key space over the previous studies' comparison, as represented in Table 6.

V. Conclusion

This paper proposes a novel dual-parallel processing dynamic gate network CNN and a quantum-inspired approach for the dynamic resolution of a multi-JPEG-image joint compression encryption model, aiming to achieve better outcomes in terms of compression ratio efficiency and robust encryption performance. The implemented framework demonstrates quantum-resistant transformations with dynamic latent-fused chaos, creating a cryptographically robust system for secure image data transmission. By binding encryption keys to intrinsic image-based secure patterns through Radon variance extraction and anchoring key derivation to prime-based CRT unique solutions in latent space, the architecture establishes tamper-evident encryption. The quantum-classical synergic approach manifested through zeta-modulated phase shifts and hyperbolic curvature scaling effectively counters both classical brute-force and future quantum attacks. Empirical validation via entropy saturation and near-ideal NPCR (>99.6%) scores confirms that the latent-fused chaotic-quantum synergy produces a cipher form with negligible correlations and maximal confusion. This paradigm shift toward pattern-bound, mathematically irreversible key generation sets a foundation for post-quantum secure image transmission.

Additionally, we have future precedent in the follow-up research. Because our work is limited to arbitrary gray-scale JPEG images. And also, when a low-entropy image is used, the security of the key generation relies on the latent features of the image. A completely uniform, featureless image (e.g., a pure white screen) would produce a less complex latent vector, potentially leading to a weaker L-FC key. Although the secret seed mitigates this, it still leaves a theoretical vulnerability. In our tests, the system still encrypts such images effectively, but the entropy of the ciphertext may see a minor decrease (~7.99 to ~7.89). We need to work with different bit color images, including JPEG color with multi-scale low- to high-quality JPEG images. As we have already mentioned, the latent-fused chaotic key seed value has a future scope for anonymous mode privileges with hardware acceleration, as the vector is generated from a random value.

Acknowledgment

The authors would like to express sincere gratitude to the Department of Computer Science Engineering (CSE) at Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering & Technology (VNR VJIET), for the invaluable support and resources provided throughout this research. The facilities, academic environment, and encouragement from faculty members have significantly contributed to the completion of this work. This study would not have been possible without the institution's commitment to advancing research and innovation in the field of medical electronics.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

Our OMEN system's wallpaper application of images as datasets were analyzed during the current study. Link: https://www.hp.com/us-en/gaming-pc/omen-gaming-hub.html (after downloading the OMEN Gaming Hub application using the link, then click on the gallery option for the OMEN wallpapers, which are free to download as JPEG images only).

Author Contribution

M. Gangappa participated in preprocessing and classifying the input mechanism, providing critical feedback on the manuscript. B V V Satyanaryana contributed to the development of domain integration, the implementation of code handling, and contributed to manuscript writing and revisions. Dheeraj Avadhanula assisted with input standardization and quantum principles handling. All authors reviewed and approved the final version of the manuscript and agreed to be responsible for all aspects of the work, ensuring integrity and accuracy.

Declarations

Ethical Approval

This study was conducted by ethical standards and has received approval from the CSE department of VNR VJIET, Hyderabad, India. Informed consent was obtained from the parents or guardians of all participating students, and confidentiality and anonymity of the participants were maintained throughout the research process. All procedures adhered to ethical guidelines for research involving human subjects.

Consent for Publication Participants.

Consent for publication was given by all participants

Competing Interests

The authors declare no competing interests.

References

[1] Y. Yuan, H. He, F. Chen, and L. Qu, "On the security of JPEG image encryption with RS pairs

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

permutation," *Journal of Information Security and Applications*, vol. 82, p. 103722, Mar. 2024.

[2] H. He, Y. Yuan, Y. Ye, H.-M. Tai, and F. Chen, "Chosen plaintext attack on JPEG image encryption with adaptive key and run consistency," *Journal of Visual Communication and Image Representation*, vol. 90, p. 103733, Dec. 2022.

[3] S. J. Mousavirad and L. A. Alexandre, "Energy-aware JPEG image compression: A multi-objective approach," *Applied Soft Computing*, vol. 141, p. 110278, Apr. 2023.

[4] Z. Li *et al.*, "Nearly-lossless-to-lossy medical image compression by the optimized JPEGXT and JPEG algorithms through the anatomical regions of interest," *Biomedical Signal Processing and Control*, vol. 83, p. 104711, Feb. 2023.

[5] Z. Li *et al.*, "An optimized JPEG-XT-based algorithm for the lossy and lossless compression of 16-bit depth medical image," *Biomedical Signal Processing and Control*, vol. 64, p. 102306, Nov. 2020.

[6] I. Perfilieva and P. Hurtik, "The F-transform preprocessing for JPEG strong compression of high-resolution images," *Information Sciences*, vol. 550, pp. 221–238, Oct. 2020.

[7] G. Li, Q. Huang, W. Wang, and L. Liu, "Human visual perception-inspired medical image segmentation network with multi-feature compression," *Artificial Intelligence in Medicine*, vol. 165, p. 103133, Apr. 2025.

[8] S. Jeong, S. Jeong, S. S. Woo, and J. H. Ko, "An overhead-free region-based JPEG framework for task-driven image compression," *Pattern Recognition Letters*, vol. 165, pp. 1–8, Nov. 2022.

[9] X. Duan, B. Li, Z. Yin, X. Zhang, and B. Luo, "Robust image steganography against lossy JPEG compression based on embedding domain selection and adaptive error correction," *Expert Systems With Applications*, vol. 229, p. 120416, May 2023.

[10] I. Hussain, S. Tan, B. Li, X. Qin, D. Hussain, and J. Huang, "A novel deep learning framework for double JPEG compression detection of small size blocks," *Journal of Visual Communication and Image Representation*, vol. 80, p. 103269, Aug. 2021.

[11] M. Rahmati, F. Razzazi, and A. Behrad, "Double JPEG compression detection and localization based on convolutional auto-encoder for image content removal," *Digital Signal Processing*, vol. 123, p. 103429, Feb. 2022.

[12] W. Tan, Y. Bao, F. Meng, C. Li, and Y. Liang, "Adaptive cross-channel transformation based on self-modulation for learned image compression," *Signal Processing Image Communication*, p. 117325, Apr. 2025.

[13] T. Ma, J. Ye, Z. Xu, and J. Zhao, "Image Encryption Algorithm based on Pseudo-random Sequence," *Procedia Computer Science*, vol. 243, pp. 1231–1238, Jan. 2024.

[14] M. Li, Q. Cui, X. Wang, Y. Zhang, and Y. Xiang, "FTPE-BC: Fast thumbnail-preserving image encryption using block-churning," *Expert Systems With Applications*, vol. 255, p. 124574, Jun. 2024.

[15] Y. Yuan, H. He, Y. Yang, N. Mao, F. Chen, and M. Ali, "JPEG image encryption with grouping coefficients based on entropy coding," *Journal of Visual Communication and Image Representation*, vol. 97, p. 103975, Nov. 2023.

[16] Q. Feng *et al.*, "DHAN: Encrypted JPEG image retrieval via DCT histograms-based attention networks," *Applied Soft Computing*, vol. 133, p. 109935, Dec. 2022.

[17] W. Chen, W. Ji, Y. Wang, J. Ren, G. Sheng, and X. Hei, "Visually secure image encryption: Exploring deep learning for enhanced robustness and flexibility," *Expert Systems With Applications*, p. 126027, Dec. 2024.

[18] C. Wang and L. Song, "An image encryption scheme based on chaotic system and compressed sensing for multiple application scenarios," *Information Sciences*, vol. 642, p. 119166, May 2023.

[19] X. Li, B. Zhang, K. Wang, and Z. Li, "A multi-image encryption-then-compression scheme based on parallel compressed sensing," *Optik*, vol. 290, p. 171304, Aug. 2023.

[20] D. Huo *et al.*, "A flexible and visually meaningful multi-image compression, encryption and hiding scheme based on 2D compressive sensing," *Heliyon*, vol. 9, no. 3, p. e14072, Feb. 2023.

[21] H. Wen, Y. Huang, and Y. Lin, "High-quality color image compression-encryption using chaos and block permutation," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, p. 101660, Jul. 2023.

[22] A. Bencherqui *et al.*, "Optimal algorithm for color medical encryption and compression images based on DNA coding and a hyperchaotic system in the moments," *Engineering Science and Technology an International Journal*, vol. 50, p. 101612, Jan. 2024.

[23] K. Meng and Y. Wo, "An image compression and encryption scheme for similarity retrieval," *Signal Processing Image Communication*, vol. 119, p. 117044, Aug. 2023.

[24] Z. Zhang, Y. Cao, H. Jahanshahi, and J. Mou, "Chaotic color multi-image compression-

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

encryption/ LSB data type steganography scheme for NFT transaction security," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 10, p. 101839, Nov. 2023.

[25] B. Wang and K.-T. Lo, "Autoencoder-based joint image compression and encryption," *Journal of Information Security and Applications*, vol. 80, p. 103680, Dec. 2023.

[26] K. Mishra, S. K. Singh, and P. Nagabhushan, "Prime product encoding for lossless compression and encryption of multiple images," *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, pp. 1–7, Nov. 2024.

[27] M. Tang, G. Du, and Y.-Y. Lin, "A novel multi-color image compression encryption algorithm based on the reconstruction coefficient matrix and DNA point mutation operation," *Expert Systems With Applications*, vol. 270, p. 126620, Jan. 2025.

[28] N. Priyanka, N. Baranwal, K. N. Singh, and A. K. Singh, "YOLO-based ROI selection for joint encryption and compression of medical images with reconstruction through super-resolution network," *Future Generation Computer Systems*, vol. 150, pp. 1–9, Aug. 2023.

[29] Z. Chen, Y. Liu, G. Ke, J. Wang, W. Zhao, and S.-L. Lo, "A Region-Selective Anti-compression image encryption algorithm based on deep networks," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, May 2024.

[30] X.-D. Liu *et al.*, "Quantum image encryption algorithm based on four-dimensional chaos," *Frontiers in Physics*, vol. 12, Mar. 2024.

[31] L. Liu, C. Yin, and Y. Dong, "A quantum image encryption scheme based on quantum Arnold transform and hyper 5D chaotic system," *Journal of Applied Physics*, vol. 137, no. 16, Apr. 2025.

[32] Y.-J. Gao, H.-W. Xie, J. Zhang, and H. Zhang, "A novel quantum image encryption technique based on improved controlled alternated quantum walks and hyperchaotic system," *Physica a Statistical Mechanics and Its Applications*, vol. 598, p. 127334, Apr. 2022.

[33] R. K. Singh, B. Kumar, D. K. Shaw, and D. A. Khan, "Level by level image compression-encryption algorithm based on quantum chaos map," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 7, pp. 844–851, Jun. 2018.

[34] S. Balasubramani, P. N. Renjith, L. Kavisankar, R. Rajavel, M. Malarvel, and A. Shankar, "A quantum-enhanced artificial neural network model for efficient medical image compression," *IEEE Access*, p. 1, Jan. 2025.

[35] D. R. I. M. Setiadi, N. Rijati, A. R. Muslikh, B. V. Indriyono, and A. Sambas, "Secure image communication using Galois Field, Hyper 3D Logistic Map, and B92 Quantum Protocol," *Computers, Materials & Continua/Computers, Materials & Continua (Print)*, vol. 0, no. 0, pp. 1–10, Jan. 2024.

[36] M. Gangappa and B. V. V. Satyanarayana, "Multi-Domain steganalysis preprocessing to fusion feature for optimal stack ensemble model," *Journal of Neonatal Surgery*, vol. 14, no. 5, pp. 620–640, Apr. 2025.

[37] J. M. Borwein, D. M. Bradley, and R. E. Crandall, "Computational strategies for the Riemann zeta function," *Journal of Computational and Applied Mathematics*, vol. 121, no. 1–2, pp. 247–296, Sep. 2000.

[38] K. Zhang, Y. Zhang, P. Wang, Y. Tian, and J. Yang, "An improved SoBel Edge algorithm and FPGA implementation," *Procedia Computer Science*, vol. 131, pp. 243–248, Jan. 2018.

[39] M.-S. Hwang, W.-G. Tzeng, and W.-P. Yang, "An access control scheme based on Chinese remainder theorem and time stamp concept," *Computers & Security*, vol. 15, no. 1, pp. 73–81, Jan. 1996.

[40] D. Li, J.-A. Lu, X. Wu, and G. Chen, "Estimating the bounds for the Lorenz family of chaotic systems☆," *Chaos Solitons & Fractals*, vol. 23, no. 2, pp. 529–534, Jun. 2004.

[41] H. H. Barrett, "III The Radon Transform and its applications," in *Progress in optics*, 1984, pp. 217–286.

[42] M. Barbosa, A. Moss, D. Page, Departamento de Inform´atica, Universidade do Minho, and Department of Computer Science, University of Bristol, "Compiler Assisted Elliptic Curve Cryptography," journal-article, 2005.

[43] M. Düll *et al.*, "High-speed Curve25519 on 8-bit, 16-bit, and 32-bit microcontrollers," *Designs Codes and Cryptography*, vol. 77, no. 2–3, pp. 493–514, May 2015.

[44] J. Lee, D. Kim, and S. C. Seo, "Parallel implementation of GCM on GPUs," *ICT Express*, Feb. 2025.

[45] S. E. S. Castelo, R. J. L. Apostol IV, D. M. A. Cortez, R. M. Dioses, M. C. R. Blanco, and V. A. Agustin, "Modification of SHA-512 using Bcrypt and salt for secure email hashing," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, p. 398, Jan. 2024.

[46] U. Garg, "Wavelet transform domain for deep image compression using high frequency Sub-Band prediction," *Türk Bilgisayar Ve Matematik*

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

Eğitimi Dergisi, vol. 10, no. 1, pp. 612–617, Apr. 2019.

[47] H. Zhu, R. Wang, Y. Jin, K. Liang, and J. Ning, "Distributed additive encryption and quantization for privacy preserving federated deep learning," *Neurocomputing*, vol. 463, pp. 309–327, Aug. 2021.

[48] D. Gowda V. *et al.*, "A novel method of data compression using ROI for biomedical 2D images," *Measurement Sensors*, vol. 24, p. 100439, Aug. 2022.

[49] Y. S. Alslman, A. Ahmad, and Y. AbuHour, "Enhanced and authenticated cipher block chaining mode," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 4, pp. 2357–2362, Mar. 2023.

[50] S. Suhaili, N. Julai, R. Sapawi, and N. Rajaee, "Towards Maximising Hardware Resources and Design Efficiency via High-Speed Implementation of HMAC based on SHA-256 Design," *Pertanika Journal of Science & Technology*, vol. 32, no. 1, Nov. 2023.

[51] X. An, S. Liu, L. Xiong, J. Zhang, and X. Li, "Mixed gray-color images encryption algorithm based on a memristor chaotic system and 2D compression sensing," *Expert Systems With Applications*, vol. 243, p. 122899, Dec. 2023.

[52] Y. Deng, J. Chen, and J. Wang, "An image compression encryption based on the semi-tensor product and the DFT measurement matrix," *Optik*, vol. 288, p. 171175, Jul. 2023.

[53] B. Jin, L. Fan, B. Zhang, R. Lei, and L. Liu, "Image encryption hiding algorithm based on Digital Time-Varying Delay Chaos Model and Compression sensing technique," *iScience*, vol. 27, no. 9, p. 110717, Aug. 2024.

[54] X. Jiang *et al.*, "Reservoir computing based encryption-then-compression scheme of image achieving lossless compression," *Expert Systems With Applications*, vol. 256, p. 124913, Jul. 2024.

[55] Y. Lu, M. Gong, L. Cao, Z. Gan, X. Chai, and A. Li, "Exploiting 3D fractal cube and chaos for effective multi-image compression and encryption," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 3, pp. 37–58, Feb. 2023.

[56] Y. Lu, M. Gong, Z. Huang, J. Zhang, X. Chai, and C. Zhou, "Exploiting compressed sensing (CS) and RNA operations for effective content-adaptive image compression and encryption," *Optik*, vol. 263, p. 169357, May 2022.

[57] K. N. Singh, O. P. Singh, and A. K. Singh, "ECiS: Encryption prior to compression for digital image security with reduced memory," *Computer Communications*, vol. 193, pp. 410–417, Aug. 2022.

[58] H. Wen, Y. Huang, and Y. Lin, "High-quality color image compression-encryption using chaos and block permutation," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, p. 101660, Jul. 2023.

[59] H. Zhang, S.-X. Nan, Z.-H. Liu, J. Yang, and X.-F. Feng, "Lossless and lossy remote sensing image encryption-compression algorithm based on DeepLabv3+ and 2D CS," *Applied Soft Computing*, vol. 159, p. 111693, May 2024.

[60] L. Zhu, D. Jiang, J. Ni, X. Wang, X. Rong, and M. Ahmad, "A visually secure image encryption scheme using adaptive-thresholding sparsification compression sensing model and newly-designed memristive chaotic map," *Information Sciences*, vol. 607, pp. 1001–1022, Jun. 2022.

Author Biography

Dr. Malige Gangappa is an Associate Professor in the Department of Computer Science and Engineering at VNR VJIET. He has been recognized by the institute multiple times for his outstanding teaching excellence. With expertise in Machine Learning, R Programming, and Python Programming, he has delivered lectures at various organizations. Dr. Gangappa was honoured with the **Best Teaching Faculty** and **Best Researcher Award** at the International Award Ceremony on Star Achievers in Engineering, Management, Arts, and Science (SAEMAS - 2022). Additionally, he is the author of books on **Computer Architecture and Object-Oriented Programming through Java**. He has published numerous research papers in reputed national and international journals. His research interests include **Deep Learning, Video Processing, and Soft Computing.**

Balla V V Satyanarayana is a passionate computer science enthusiast currently pursuing a Master of Technology in Computer Science and Engineering at VNR VJIET. He completed his undergraduate studies in the same field. He gained over a year of professional experience as a **Full-Stack Web Developer**, with a strong focus on backend development using **Java and Network Management** in **Cybersecurity**. His industry background fuels his academic curiosity, particularly in the emerging intersection of **Cybernetics and Deep learning**. His

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156

current research explores how quantum-level mapping can enhance intelligent systems and secure networks. Always eager to bridge practical experience with cutting-edge innovation, he aims to contribute to the evolving field of **AI and Quantum Computing**.

Dheeraj A completed his undergraduate studies in Computer Science and Engineering and has pursued a Master's degree in CSE from the University of North Texas. With a strong interest in systems and intelligent computing, his research focuses on applying **Artificial Intelligence to Operating System Kernel Management**. He is particularly curious about how AI can enhance system efficiency, automate resource allocation, and improve kernel-level decision-making. Combining a solid foundation in computer science with a forward-thinking approach, he aims to contribute to the evolution of more intelligent, more adaptive operating systems.

Manuscript received 20 July 2025; Revised 10 September 2025; Accepted 2 October 2025; Available online 6 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.1156