



OPEN

A hybrid encryption framework leveraging quantum and classical cryptography for secure transmission of medical images in IoT-based telemedicine networks

Arslan Shafique^{1✉}, Syed Ali Atif Naqvi¹, Ali Raza¹, Masoud Ghalaii², Panagiotis Papanastasiou³, Julie McCann⁴, Qammer H. Abbasi¹ & Muhammad Ali Imran¹

In the era of the Internet of Things (IoT), the transmission of medical reports in the form of scan images for collaborative diagnosis is vital for any telemedicine network. In this context, ensuring secure transmission and communication is necessary to protect medical data to maintain privacy. To address such privacy concerns and secure medical images against cyberattacks, this research presents a robust hybrid encryption framework that integrates quantum and classical cryptographic methods. The proposed framework not only secure medical data against cyber threats but also protects the secret security keys. Initially, a Quantum Key Distribution (QKD) is employed to generate a shared key, which is then used to secure the symmetric keys via One-Time Pad (OTP) encryption. Next, bit-planes are extracted from each color component. The rows and columns of the extracted bit-planes are scrambled using random sequences which are generated by a 6D hyperchaotic Chen system and the Ikeda map. To further increase confusion in the original data, multiple-step pixel scrambling operations such as pixel shuffling, pixel value shuffling, and rotational and flipping operations are implemented. After the confusion phase, a combination of affine transformations with non-linear functions, Discrete Cosine Transform (DCT) with complex modulation, Discrete Wavelet Transform (DWT) with random phase modulation, bilinear transformation, and nonlinear polynomial mapping are employed to create diffusion in the scrambled components. These multiple encryption operations aim to maximize randomness in the final ciphertext image. Additionally, to reduce computational complexity, only the Most Significant Bit-Planes (MSBs) are encrypted, as they contain more than 94% of the plaintext information. Several experimental results and analyses are conducted to assess the proposed encryption framework, including entropy analysis, key sensitivity analysis, correlation analysis lossless analysis, and histogram analysis. Furthermore, the framework is tested against various cyberattacks such as brute-force attacks, clipping attacks, and noise attacks on the ciphertext images, to demonstrate its resilience against such threats.

The evolution of the Internet from digitization to intellectualization is transforming daily life and industry¹. As the transfer of digital data, specifically digital images, occurs in huge amounts between Internet of Things (IoT) devices, it is necessary to secure them from eavesdropper access. Developing a robust technology to secure digital images is not enough because IoT devices such as smartphones and tablets are very resource-constrained, including limited power and limited storage². As digital images contain a huge amount of data with a high correlation between image pixels, while developing algorithms for securing digital data, it should be kept in mind that they should be capitalised with such IoT devices having limited resources.

To protect image information from unauthorised access, three main technologies, such as encryption methods and steganographic techniques, and techniques for securing encryption secret keys have been frequently used in the past several years^{3–5}. Image encryption includes several approaches such as frequency

¹School of Electronic and Nanoscale Engineering, University of Glasgow, Glasgow G12 8QQ, UK. ²School of Electronic and Electrical Engineering, University of Leeds, Leeds LS2 9JT, UK. ³Department of Computer Science, University of York, York YO10 5DD, UK. ⁴Department of Computing, Imperial College, London SW7 2AZ, UK. ✉email: Arslan.shafique@glasgow.ac.uk

domain encryption⁶, spatial domain encryption, compressed sensing⁷, and encryption based on optical and quantum computing^{8,9}. Spatial domain encryption modifies pixel values directly, while transform-based encryption uses transformations to alter pixel values in both the spatial and transform domains. Compressive sensing incorporates both compression and encryption to enhance the processing time while also increasing the chances of losing data due to decryption. A new image encryption approach that enhances the use of encryption with optical properties¹⁰. Quantum image encryption, on the other hand, employs quantum encoders to secure digital images from cyber threats¹¹. It is worth noting that encryption methods based on quantum phenomena are rather superior to those classical methods, which are logistic in nature. Most of the encryption techniques have a positive impact depending on the purpose of use^{12–15}. For example, the advanced encryption standard (AES)¹⁶ is very effective when enhanced privacy is needed over digital pictures, but when it comes to IoT devices, it is not suitable as it has multiple encryption rounds and also includes very high-resolution encrypted images that take large storage space. However, with the growth in quantum computing, even the most secure algorithms like AES could become vulnerable.

To overcome the vulnerabilities related to the high latency and weak security, quantum operations such as quantum scrambling, and quantum substitution can be utilized. The scrambling process rearranges positions of the image pixel using random numbers generated by methods such as Blum Blum Shub¹⁷, chaotic systems^{7,11,18}, ChaCha20¹⁹, and Random Number Generation Uniform (RANDU)²⁰. Among them, chaotic systems are very reliable due to their robust properties, such as sensitivity to initial conditions, aperiodicity, topological mixing, and ergodicity. However, pixel permutation alone does not provide strong security for images, as it only rearranges pixel positions without altering the pixel values themselves²¹. Combining scrambling with diffusion provides stronger encryption by reducing the image pixel correlations. An effective encryption should use both permutation and diffusion for strong protection. Traditional encryption methods such as AES, and DES are unsuitable for real-time applications like video conferencing due to limited efficiency and security. However, quantum image encryption provides enhanced security and efficiency by leveraging quantum mechanics.

In this research, a robust and time-efficient hybrid quantum encryption framework is proposed. The proposed encryption framework utilizes QKD for key management, where a shared key is generated through QKD and then encrypted using a One-time Pad (OTP). This combination ensures that the secret keys are highly secure which makes it virtually impossible for eavesdroppers to recover the original secret keys. Moreover, the encryption process begins by extracting the R, G, and B components from the plaintext image. Bit-planes are then extracted from each color component of the input image. As most of the plaintext information present in the Most Significant Bit Planes (MSBPs)²², therefore, only these bit-planes are selected for encryption to reduce the computational complexity of the proposed encryption framework. The remaining, unencrypted bit-planes are simply combined to form pre-ciphered images for each color component. For the encryption of the MSBPs, random sequences are generated using specific initial conditions and control parameters of the chaotic maps such as 6D hyper Chen chaotic map (6DHCCM), and Ikeda map that act as secret keys. These sequences are employed to perform three types of scrambling: pixel scrambling, pixel value permutation, and rotation and flipping. Additionally, the process of generating random sequences is extended to enhance the diffusion process using XOR operation. Further diffusion is achieved by applying a variety of operations, including Affine transformations with nonlinear functions, discrete cosine transform (DCT), wavelet transform with random phase modulation, bilinear transformation with complex weights, and nonlinear polynomial mapping to produce the final encrypted image. The contributions of the paper are as follows:

- A novel hybrid encryption framework is proposed that combines quantum and classical cryptographic techniques developed to secure the transmission of medical images in IoT-based telemedicine networks.
- In the proposed work, QKD is combined with OTP encryption to generate shared keys and secure the symmetric secret keys, which enhances the key management process.
- To reduce the computational complexity, bit-planes are extracted from the color components of the medical images and scramble only such bit-planes that contain the majority of the image information (over 94%) using random sequences generated by a 6D hyperchaotic Chen system and the Ikeda map.
- Instead of using a single scrambling step, three-step pixel scrambling operations such as pixel shuffling, pixel value shuffling, and rotational and flipping operations are applied to create randomness in the plaintext image.
- In the diffusion phase, affine transformations with non-linear functions, DCT with complex modulation, DWT with random phase modulation, bilinear transformations, and nonlinear polynomial mapping are combined to maximize randomness in the encrypted images. The remainder of the paper is structured as follows: Section 2 offers a brief summary of existing works including their advantages and vulnerabilities. Also, this section provides the advantages, vulnerabilities, and possible solutions to overcome such vulnerabilities. Section 3 delivers a comprehensive explanation of the foundational knowledge necessary for the development of the proposed encryption framework. Section 4 presents a step-by-step explanation of the proposed encryption framework. In Section 5, the experimental results and analysis of the proposed encryption framework are discussed. Section 6 presents a concise discussion of the entire proposed work, along with an overview of the advantages of utilizing multiple encryption layers within the framework. Finally, Section 7 concludes the entire proposed research, highlights the limitations of the proposed work, and also provides a few recommendations for future research to address these limitations.

Related work

The development of image encryption techniques has seen substantial progress, incorporating contemporary technologies to enhance data security. This section explores the existing encryption schemes and investigates their vulnerabilities. For instance, In²³, Hu et al. proposed a quantum image encryption method using a qubit-level scrambling^{23,24}. This approach enhances operational effectiveness by the use of hyperchaotic states and

sometimes controlling periodic windows, while the diffusion and scrambling methods aid image encryption further. Quite, the scheme is not effective if the parameters of the chaotic maps and quantum model are not efficiently selected.

In²⁵, He et al. proposed a quantum image encryption algorithm, that enhances quantum image preparation by optimizing quantum circuits to reduce auxiliary qubits. The algorithm also deals with non-random key issues with a modified logistic map²⁶ and enhances security with row-column-based permutation²⁷. However, vulnerabilities may arise if the circuit optimization or permutation methods are exploited, and the encryption could be weakened if the improved logistic map lacks adequate randomness or security. In²⁸, Mohamed et al. proposed a quantum-based framework for the encryption of digital images. They perform pixel diffusion and pixel permutation by using quantum operations like Hadamard gates²⁹ and quantum-controlled gates³⁰ and perform key generation using a hyperchaotic system, bloomschaotic sequences, and circular shifts for permutation. This scheme poses a risk of attacks owing to the hyperchaotic being predictable. In³¹, Gao et al. proposed quantum DNA-based cryptographic techniques. To round off the pixel transformation images of the new frame, alternation of colours of particular pixels within the images is achieved using Hilbert scrambling³⁰. Their scheme is vulnerable to attacks due to weak management of the inverse operations and the key matrix. In³², Zhang et al. presented quantum image cryptosystems that include superposition states in reversible pixel scramblers. It also incorporates a pixel reordering feature in which a feature-orientated order of the image is processed with sequential cries. In³³, Wen et al presented an encryption scheme based on chaos-based block scrambling and confusion-diffusion operations. It encrypts the digital images using multiple cryptographic operations, such as block-wise scrambling, and rotation. It also satisfies cryptographic criteria for confusion, diffusion, and avalanche effects.

In³⁴, Rehman et al. proposed an encryption framework for the protection of color image schemes dependent on OTP keys and chaos theory with a rotor machine concept. The image pixel rows and columns are transformed and rotated to create new configurations for substitution. Vulnerabilities arise in their scheme due to the predictable patterns of the rotor rotation and chaotic sequences. Moreover, the one-time keys and chaotic maps are not securely managed or sufficiently randomized. In³⁵, Kumar et al. introduced an encryption scheme that converts digital images into a 2D difference matrix using a median edge detector. The image is then encoded with a bit-plane representation to reduce the size and enable the embedding more secret data. In³⁶, Zhu et al. performed cryptanalysis of encryption frameworks using bit plane extraction to find the key streams for bit-level permutation and XOR diffusion. This independence allows attackers to recover the permutation and diffusion keys through chosen plaintext attacks using just two specific plaintext images and their ciphertexts, exposing a significant vulnerability in the encryption system. In³⁷, Sing et al. presented a chaotic system based encryption scheme in which key is initialized with confusion- diffusion operations. While the method offers strong security and high speed, it is not enough to resist cyberattacks because of the exploitable patterns in the object detection process. A summary of the existing works is provided in Table 1.

Preliminaries

In the proposed research, several key components are used, and it is essential to understand their importance and roles within the framework. The following subsections provides a brief overview of these critical encryption elements and explain why they are integral to the proposed encryption approach.

6D Hyperchaotic Chen system

The 6D Hyperchaotic Chen system (6DHCCS)³⁸ is a higher-dimensional extension of the classic Chen system which is designed to exhibit complex and extremely chaotic behavior. The additional dimensions in this system increase its complexity and unpredictability which makes it suitable for secure communications and cryptography applications. Mathematically, 6DHCCS can be expressed using Equation 1.

Existing schemes	Application domain	Real-World performance	Robustness against attacks	disadvantages	Potential solutions
Hu et al. ²³	Quantum Image Encryption	Enhanced security	Vulnerable to poor parameters	Chaotic maps vulnerabilities	Improve parameter selection
He et al. ²⁵	OCBPB Quantum Encryption	Optimized BRQI prep	Circuit and key randomness issues	Weak circuit optimization	Better optimization techniques
Mohamed et al. ²⁸	Quantum Cellular Automata	Effective diffusion	Predictable hyperchaotic system	Predictable chaos	Use less predictable chaos
Gao et al. ³¹	Quantum DNA Code	Uses Hilbert scrambling	Weak key management	Weak key matrix	Improve key management
Zhang et al. ³²	Quantum State Superposition	Large key space	Vulnerable to attacks	Weak state transmission	Enhance state security
Wen et al. ³³	Chaos-Based Block Permutation	Satisfies criteria	Not detailed	Not addressed	Further robustness evaluation
Rehman et al. ³⁴	Color Image Encryption	Transforms pixel rows	Predictable rotor patterns	Weak key management	Secure key management
Kumar et al. ³⁵	Bit-Plane Representation	Effective embedding	Not detailed	Limited robustness details	Enhance robustness evaluation
Zhu et al. ³⁶	Bit Plane Extraction	Key streams analyzed	Vulnerable to attacks	Exploitable key recovery	Improve key management
Singh et al. ³⁷	Chaos-Based for Smart Cities	Fast object detection	Exploitable detection patterns	Patterns in detection	Better detection security

Table 1. Summary of Image Encryption Schemes.

$$\begin{aligned}
\dot{\varpi}_1 &= \sigma_1(\varpi_2 - \varpi_1) \\
\dot{\varpi}_2 &= \rho_1 \varpi_1 - \varpi_2 - \varpi_1 \varpi_3 \\
\dot{\varpi}_3 &= -\beta_1 \varpi_3 + \varpi_1 \varpi_2 \\
\dot{\varpi}_4 &= \sigma_2(\varpi_5 - \varpi_4) \\
\dot{\varpi}_5 &= \rho_2 \varpi_4 - \varpi_5 - \varpi_4 \varpi_6 \\
\dot{\varpi}_6 &= -\beta_2 \varpi_6 + \varpi_4 \varpi_5
\end{aligned} \tag{1}$$

where $\varpi_1, \varpi_2, \varpi_3, \varpi_4, \varpi_5, \varpi_6$ are the state variables, and $\sigma_1, \rho_1, \beta_1, \sigma_2, \rho_2, \beta_2$ are the system parameters. In 6D, visualization can be challenging, but projections of the attractor into lower-dimensional spaces (such as 3D) reveal intricate and complex trajectories as shown in Figure 1. By analyzing various 3D projections of the system—such as $\varpi_1 - \varpi_2 - \varpi_3, \varpi_4 - \varpi_5 - \varpi_6$, and combinations involving other dimensions, it can be seen that each projection reveals unique patterns and interactions among the state variables, which can be utilized to generate diverse and unpredictable pseudorandom sequences for encrypting image pixels. Moreover, 3D projections of the 6D hyperchaotic Chen system highlight its complex trajectories and high sensitivity, which enable it to disrupt predictable image information when used in pixel scrambling.

The values of the state variables $\varpi_1, \varpi_2, \varpi_3, \varpi_4, \varpi_5, \varpi_6$ are obtained through numerical integration of the 6D Hyperchaotic Chen System's differential equations, and these values are computed over time and are not fixed but rather evolve according to the system dynamics. However, the values of $\sigma_1, \rho_1, \beta_1, \sigma_2, \rho_2, \beta_2$ are 70, 150, 20, 70, 150, and 20, respectively.

Ikeda map

The Ikeda map is a discrete dynamical system that models the behavior of light in an optical cavity with a nonlinear medium³⁹. Equation 2 is the mathematical representation of Ikeda map.

$$\begin{aligned}
\Xi_{n+1} &= 1 + u(x_n \cos(\theta_n) - \xi_n \sin(\theta_n)) \\
\xi_{n+1} &= u(\Xi_n \sin(\theta_n) + \xi_n \cos(\theta_n))
\end{aligned} \tag{2}$$

where: Ξ_n and ξ_n are the coordinates of the point at the n^{th} iteration. u controls the behavior of the map, and θ_n represents the phase shift at the n -th iteration as given in Equation 3.

$$\theta_n = k - \frac{p}{1 + x_n^2 + y_n^2} \tag{3}$$

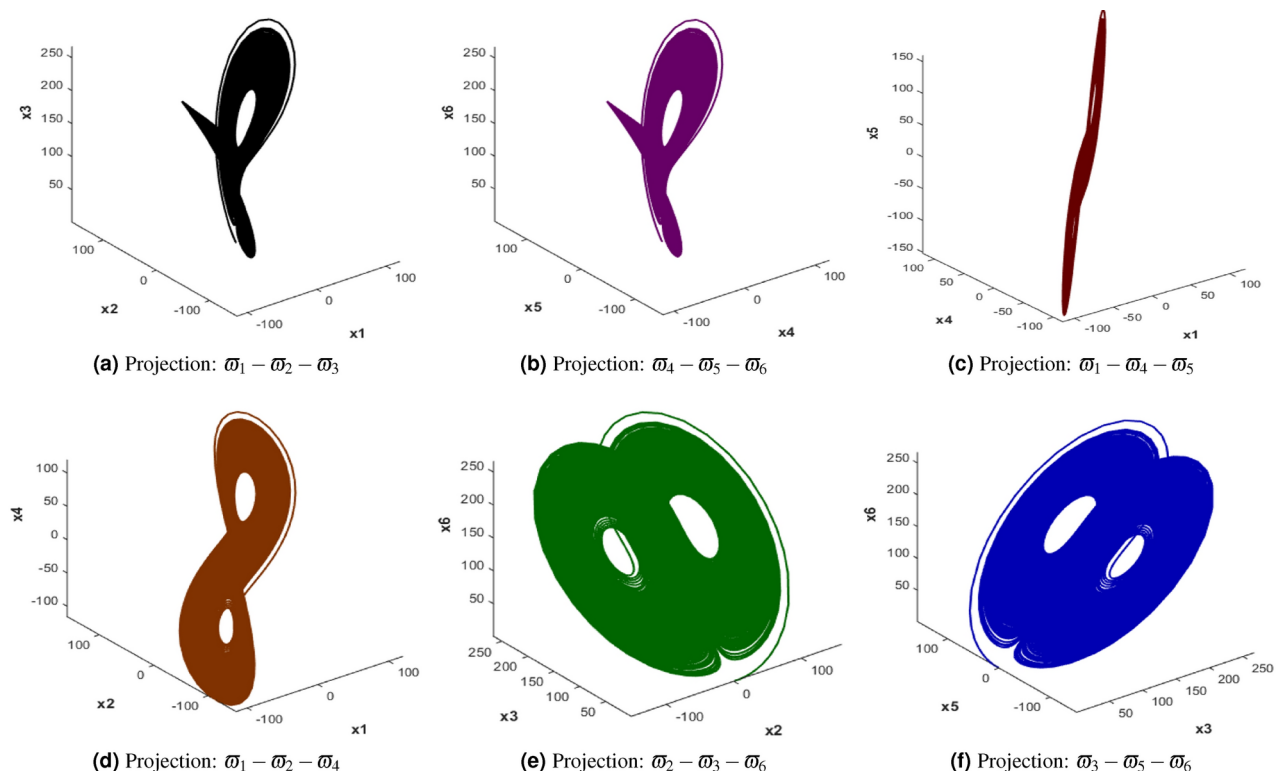


Fig. 1. 6D Hyperchaotic Chen system attractors.

where k and p control the nonlinearity and feedback in the system. Figure 2(a-f) illustrates the six attractors of Ikeda map attractors under different initial conditions that demonstrate its chaotic behavior and sensitivity, which produce non-repeating sequences for further scrambling and diffusion of image data. Together, these chaotic systems can be used to create strong randomness in the encrypted image to enhance security by concealing any identifiable structure and making it difficult for attackers to retrieve the original image.

Proposed encryption process

The proposed encryption process includes four main phases: (i) Generating a shared key using QKD with OTP encryption, (ii) extracting bit-planes from each color component and encrypting them through scrambling operations using random sequences generated by the 6DHCCS and the Ikeda map, (iii) applying confusion operations through multiple permutation techniques, and (iv) achieving diffusion in the scrambled image. A detailed overview of the entire encryption framework is illustrated in Figure 3. Each step is explained in detail in the following subsections.

The proposed encryption process consists of four key phases: (i) Generating of the shared key using QKD which is further based on OTP encryption chaotic systems, (ii) bit-planes extraction from each color component, and encrypting using scrambling operations applied based on the random sequences generated using 6DHCCS, and Ikeda map, (iii) carrying out confusion operations using multiple permutation operations, and (iv) creating diffusion in the scrambled image. The details of each step are provided in the next subsections.

QKD based symmetric key sharing

Our proposed encryption framework begins with the implementation of QKD for generating secret keys. QKD allows multiple parties to generate shared secret cryptographic keys, which can be used for secure communication. In the proposed research, the BB84 algorithm, which is proposed by Charles Bennett and Gilles Brassard⁴⁰, is used to secure the actual symmetric keys that are used in the generation of the chaotic sequences.

Alice randomly prepares a sequence of quantum bits (qubits), where each qubit can be in one of four possible states, based on two different bases: Rectilinear (+) and Diagonal ($\times$). The rectilinear (+) and the diagonal ($\times$) basis states are $|1\rangle$ (bit value 1), $|1\rangle$ (bit value 1) and $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ (bit value 0), $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ (bit value 1), respectively.

Now Alice prepares a sequence of 8 qubits. For this example, bit sequence (b) is (1, 0, 1, 1, 0, 1, 0, 0), and basis choice (b_B) are (+, $\times$, +, $\times$, +, $\times$, +, $\times$). Here, the Rectilinear basis + is represented as $|0\rangle$ and $|1\rangle$, and the Diagonal basis $\times$ is represented as $|+\rangle$ and $|-\rangle$.

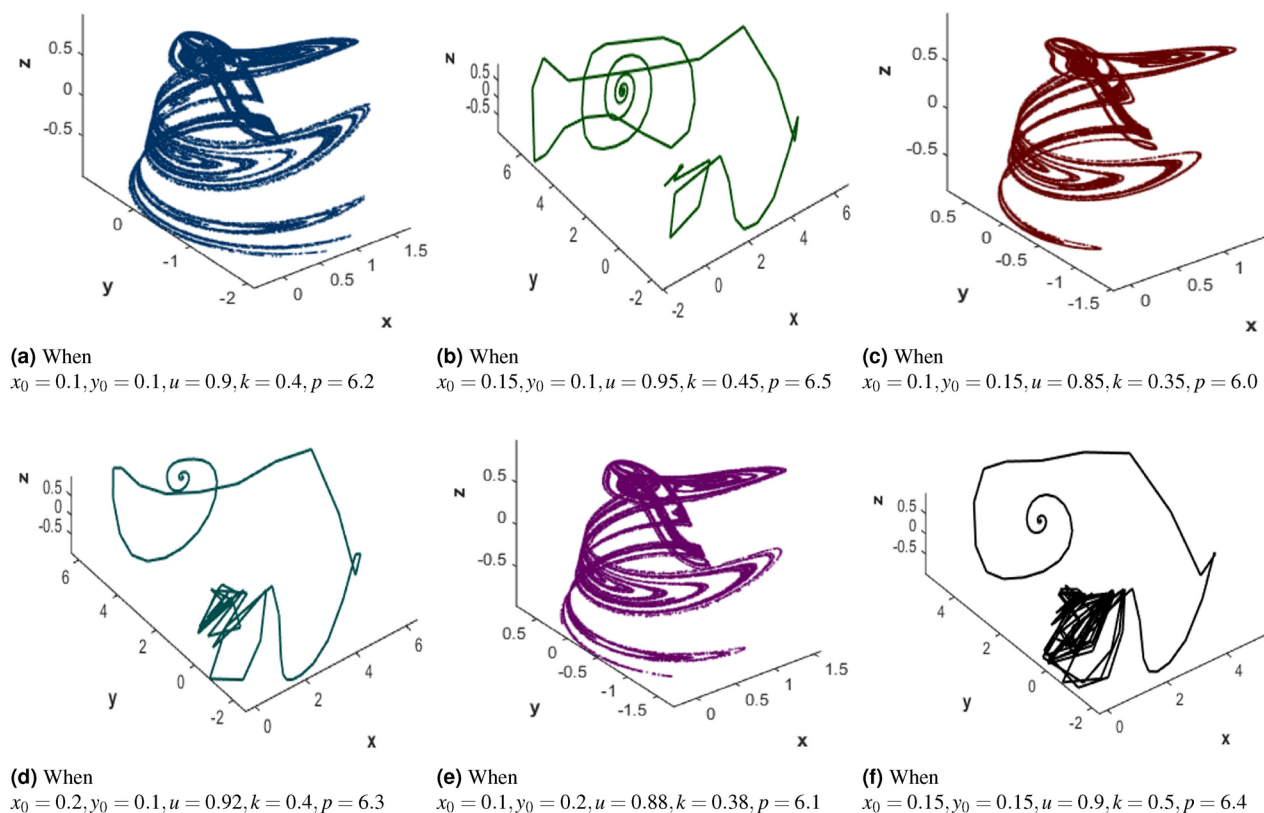


Fig. 2. 3D Ikeda map attractors demonstrating chaos and sensitivity.

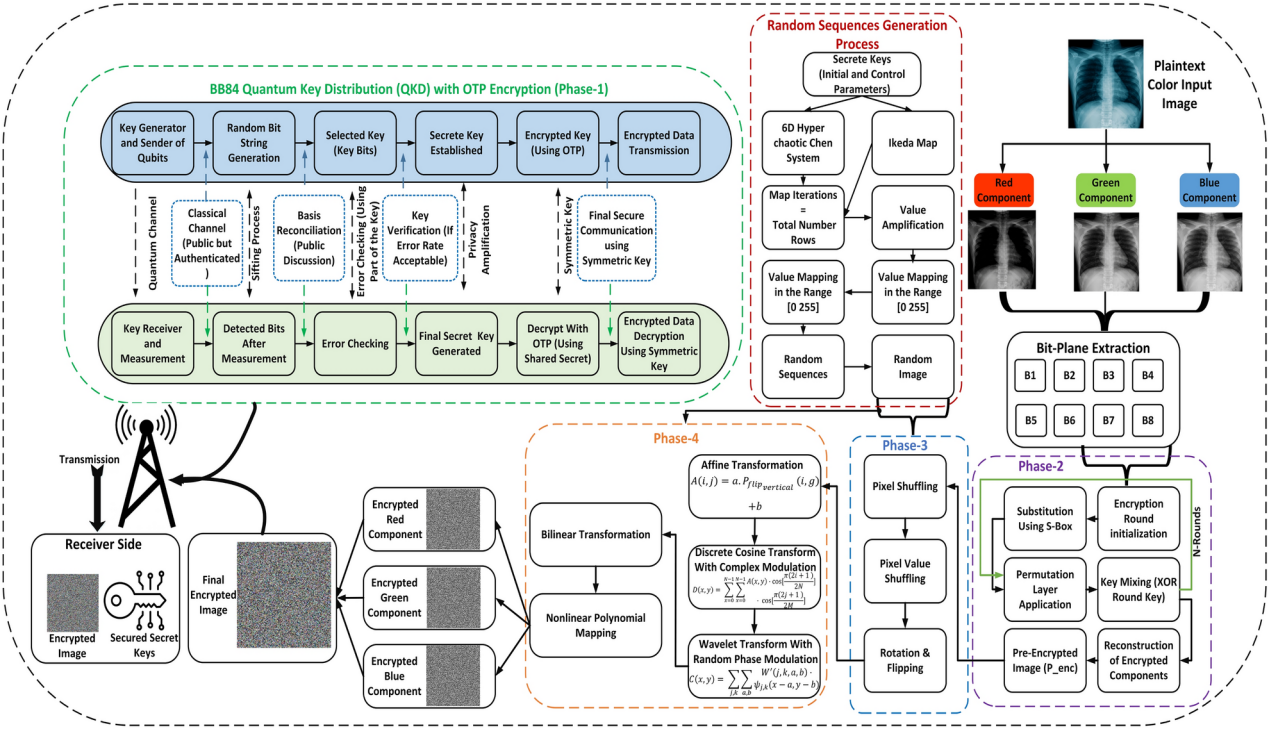


Figure 3. Propped encryption framework.

Bits	Qubits	Measured results
1.	Qubit 1	Measured in $\times$ basis, Bob gets 1 (correct basis)
2.	Qubit 2	Measured in $+$ basis, Bob gets 0 (correct basis)
3.	Qubit 3	Measured in $\times$ basis, Bob gets nothing (incorrect basis)
4.	Qubit 4	Measured in $+$ basis, Bob gets 0 (correct basis)
5.	Qubit 5	Measured in $\times$ basis, Bob gets 0 (correct basis)
6.	Qubit 6	Measured in $+$ basis, Bob gets 1 (correct basis)
7.	Qubit 7	Measured in $\times$ basis, Bob gets 1 (correct basis)
8.	Qubit 8	Measured in $\times$ basis, Bob gets nothing (incorrect basis)

Table 2. Bob’s basis choice and measurement.

Bob randomly chooses his measurement basis. For instance, Bob’s basis ($b_{B'}$) are $(\times, +, +, \times, \times +, \times, +)$. Now Bob will measure the qubits and obtain the following results as mentioned in Table 2. Now Alice and Bob publicly compare their chosen bases and retain only those bits where their bases match.

- **Alice’s Bases:** $(+, \times, +, \times, +, \times, +, \times)$
- **Bob’s Bases:** $(\times, +, \times, \times, +, \times, +, +)$

The retained bits will be 1,0,0,0,1,1. Therefore, Alice and Bob both have the shared key of 1,0,0,0,1,1. No, the following procedure is used to encrypt the symmetric key.

- **Define the symmetric key:** Assume $\alpha = 3.49$. Convert α to binary, which is $\alpha_{\text{binary}} = 11.011111$. Suppose a scaling factor of 100 is used (two decimal places), $3.49 \times 100 = 349 \rightarrow -349$ in binary is 101011101₂. Now pad the binary of 100011 to match the length of 101011101:

$$\begin{array}{cccccccccc} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & \\ \hline 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & \end{array}$$

This merged output (M) is sent over the public channel. The receiver, knowing both the shared key and the merged output, performs XOR again to recover the original symmetric key α . This ensures that even if the merged output is intercepted by the eavesdropper, without the shared key, it is impossible to recover the symmetric key.

Input: Color image I , S-box S-box, Round keys K , Number of rounds n

Output: Pre-encrypted image I_{P-enc}

Initialize R_{enc} , G_{enc} , B_{enc} as zero matrices of size (height $\times$ width)

for each color channel c in $\{1, 2, 3\}$ **do**

for $k = 0$ to 7 **do**

 Extract bit-plane $P_{C,k}$ from channel c

$P_{C,k} \leftarrow \text{ApplyEncryption}(P_{C,k}, \text{S-box}, K, n)$

if $c = 1$ **then**

$R_{enc} += P_{C,k} \cdot 2^k$

else if $c = 2$ **then**

$G_{enc} += P_{C,k} \cdot 2^k$

else if $c = 3$ **then**

$B_{enc} += P_{C,k} \cdot 2^k$

end if

end for

end for

Combine R_{enc} , G_{enc} , B_{enc} to form I_{P-enc} **return** I_{P-enc}

Function ApplyEncryption(P_C_k, S-box, K, n):

Input: Bit-plane $P_{C,k}$, S-box S-box, Round keys K , Number of rounds n

Output: Encrypted bit-plane $P_{C,k}$

for $i = 1$ to n **do**

$P_{C,k} \leftarrow \text{S-box}(P_{C,k})$

$P_{C,k} \leftarrow \text{Permutation}(P_{C,k})$

$P_{C,k} \leftarrow P_{C,k} \oplus K_i$

end for

return $P_{C,k}$

Function Permutation(P_C_k):

Input: Bit-plane $P_{C,k}$

Output: Permuted bit-plane $P_{C,k}$

Implement permutation logic here

return P_{enc}

▷ Bit-plane

Algorithm 1. Bit-plane extraction based permutation.

- 1: **Input:** Image $P_{enc}(M \times N)$
- 2: **Pixel Shuffling:**
- 3: Define permutation functions π_r for rows and π_c for columns
- 4: **Row Scrambling:**
- 5: $P_{shuffled_rows}(i, j) \leftarrow P_{enc}(\pi_r^{-1}(i), j)$
- 6: **Column Scrambling:**
- 7: $P_{shuffled}(i, j) \leftarrow P_{shuffled_rows}(i, \pi_c^{-1}(j))$
- 8: **Pixel Value Permutation:**
- 9: Flatten image $P_{shuffled}$ to vector $\mathbf{p} = [p_1, p_2, \dots, p_{MN}]$
- 10: Define permutation function π_v
- 11: $\mathbf{p}'(k) \leftarrow \mathbf{p}(\pi_v(k))$
- 12: Reshape $\mathbf{p}'$ back to $M \times N$ to get P_{perm}
- 13: **Rotation:**
- 14: Compute center $(i_c, j_c) = (\frac{M+1}{2}, \frac{N+1}{2})$
- 15:

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos \theta \cdot i - \sin \theta \cdot j + (1 - \cos \theta) \cdot i_c + \sin \theta \cdot j_c \\ \sin \theta \cdot i + \cos \theta \cdot j + (1 - \cos \theta) \cdot j_c - \sin \theta \cdot i_c \end{bmatrix}$$

- 16: **Horizontal Flipping:**
- 17: $P_{flip_horizontal}(i, j) \leftarrow P_{perm}(i, N - j + 1)$
- 18: **Vertical Flipping:**
- 19: $P_{flip_vertical}(i, j) \leftarrow P_{flip_horizontal}(M - i + 1, j)$
- 20: **Output:** Encrypted image $P_{flip_vertical}$

Algorithm 2. Confusion phase.

Justification of secret key security

The security of the process relies on the properties of the XOR operation and the fact that the shared key is known only to the legitimate parties (Alice and Bob).

The XOR operation has the property that if you know the result $C = A \oplus B$ and one of the operands (either A), one can uniquely determine the other operand ($B = C \oplus A$). However, if you only know C without knowing either A or B , it is computationally infeasible to determine either A or B without additional information.

In the proposed setup, the shared key K is known only to Alice and Bob, and it is never shared publicly. Even if an attacker (Eve) intercepts the merged output, without knowledge of the shared key K , the receiver cannot recover the symmetric key α .

Mathematical proof of security

Theorem 1 Prove that an attacker who intercepts the merged output $M = \alpha \oplus K$ cannot recover α without knowing K .

- **Given:**
- Symmetric key: α
- Shared key: K
- Merged output: $M = \alpha \oplus K$
- **To prove:** Knowing M alone does not allow an attacker to recover α .

Case 1: Eve has only the merged output $M = \alpha \oplus K$. To recover α , Eve needs to compute: $\alpha = M \oplus K$.

Challenge for Eve: Eve knows M , but M depends on both α and K . K is unknown to Eve. Therefore, the XOR operation is such that every possible key K could correspond to a different symmetric key α . This means that without knowing K , Eve cannot uniquely determine α .

Case 2: Randomness of K : Assume K is a random string of bits, generated independently of α . Since K is random and known only to Alice and Bob, the merged output M can be any random binary sequence of the same length as α . Therefore, for every possible value of M , there is an equally likely K that could correspond to any possible α . Thus, from M alone, the probability distribution of possible values of α is uniform over all possible values, providing no information about α .

Case 3: Conditional Entropy: The conditional entropy $H(\alpha|M)$ represents the uncertainty about α given M . Since K is random and unknown to Eve, the conditional entropy remains high: $H(\alpha|M) = H(\alpha)$. This means that knowing M does not reduce the uncertainty about α .

Image encryption

In the proposed encryption scheme following major components are used to provide robust security to the digital images.

- Bit-plane extraction-based cryptography
- Random numbers generated using chaos
- Confusion phase: Multiple scrambling operations
- Diffusion phase: XOR operation, Affine transformations with nonlinear functions, discrete cosine transform (DCT), wavelet transform with Random Phase Modulation, bilinear transformation with complex weights, and nonlinear polynomial mapping

Bit-plane extraction based cryptography

Let I be the input color image, which can be represented as a 3D matrix $I(x, y, c)$ where x and y are the spatial coordinates, and c denotes the color channel (Red, Green, Blue) as given in Equation 4.

$$I(x, y, c) = \begin{cases} R(x, y) & \text{if } c = 1 \\ G(x, y) & \text{if } c = 2 \\ B(x, y) & \text{if } c = 3 \end{cases} \quad (4)$$

For each color component $C(x, y) \in \{R(x, y), G(x, y), B(x, y)\}$, extract the k -th bit-plane $P_{C,k}(x, y)$ using Equation 5.

$$P_{C,k}(x, y) = \left\lfloor \frac{C(x, y)}{2^k} \right\rfloor \bmod 2 \quad (5)$$

where k ranges from 0 to 7, representing each bit-plane. Apply encryption functions as given in Equations 6–10 to each bit-plane extracted from the R, G, and B components. Suppose, there are n rounds of encryption with a round key K_i for each round i .

Now, use a substitution function S (S-box) for non-linear substitution using Equation 6. The S-box used in this research are given in^{41–43}.

$$S(P_{C,k}(x, y)) = \text{S-box}(P_{C,k}(x, y)) \quad (6)$$

The permutation layer, which scrambles the bits, and the key mixing, which XORs the result with the round key, are applied according to Equation 7.

$$\begin{cases} P'_{C,k}(x, y) = P(S(P_{C,k}(x, y))) \\ P''_{C,k}(x, y) = P'_{C,k}(x, y) \oplus K_i \end{cases} \quad (7)$$

Repeat substitution, permutation, and key mixing for n rounds. Let $P_{C,k}^n(x, y)$ be the pre-encrypted bit-plane after n rounds as given in Equation 8.

$$P_{C,k}^n(x, y) = P_n \left(S_n \left(\bigoplus_{i=1}^n (P_{C,k}^{i-1}(x, y) \oplus K_i) \right) \right) \quad (8)$$

where S_n and P_n represent the substitution and permutation operations in the n -th round.

Combine the encrypted bit-planes to create the pre-ciphertext image by reconstructing each pixel value for each color component from the encrypted bit-planes using Equation 9.

$$C_{\text{enc}}(x, y) = \sum_{k=0}^7 P_{C,k}^n(x, y) \cdot 2^k \quad (9)$$

where $C \in \{R, G, B\}$ denotes the color component. Combine the encrypted color components to form the pre-encrypted image $I_{\text{P-enc}}$ as given in Equation 10. The process to generate $P - \text{enc}$ is given in Algorithm 1.

$$P_{\text{enc}}(x, y, c) = \begin{cases} R_{\text{enc}}(x, y) & \text{if } c = 1 \\ G_{\text{enc}}(x, y) & \text{if } c = 2 \\ B_{\text{enc}}(x, y) & \text{if } c = 3 \end{cases} \quad (10)$$

Random numbers generation using chaos

To generate a random image using a chaotic map and convert it to a 2D image, the following steps are followed:

- **Choose initial values and control parameters:** Select initial values x_0 and control parameters a for the chaotic map.
- Iterate Equations 1, and 2 for $M \times N$ times to generate a sequence of values.
- **Scale the Values:** Since the chaotic map produces values in the range $(0, 1)$, multiply each value by a large integer L to amplify it as given in Equation 11.

$$y_n = x_n \cdot L \quad (11)$$

- Truncate the decimal part of y_n to convert it into an integer value.
- **Adjust to the desired range [0, 255]:** Take the modulo 256 of each value to ensure it fits within the 8-bit range according to Equation 12.

$$z_n = \text{mod}(y_n, 256) \quad (12)$$

- **Generate the image:** Reshape the 1D sequence of values into a 2D matrix with size $M \times N$, where $M \times N$ is the total number of color components in the image.

Confusion phase

In the confusion phase of the proposed encryption framework, multiple image shuffling techniques such as pixel shuffling, pixel value permutation, rotation and flipping, and scrambling by image transformation are applied to P_{enc} .

Pixel shuffling

It involves rearranging the pixels of an image according to some permutation rules. The primary goal is to disrupt the spatial structure of the image without altering the pixel values. In the proposed encryption framework, given an image P_{enc} of size $M \times N$, let each pixel be denoted as $P_{\text{enc}}(i, j)$, where $i \in \{1, 2, \dots, M\}$ and $j \in \{1, 2, \dots, N\}$. The two types of pixel shuffling such as row and column scrambling are used. For the row scrambling, a permutation function is defined as $\pi_r : \{1, 2, \dots, M\} \rightarrow \{1, 2, \dots, M\}$ that randomly shuffles the rows of the P_{enc} . The new image $P_{\text{shuffled_rows}}$ after row shuffling is given by Equation 13.

$$\begin{aligned} P_{\text{shuffled_rows}}(\pi_r(i), j) &= P_{\text{enc}}(i, j), \\ \forall i \in \{1, 2, \dots, M\}, \forall j \in \{1, 2, \dots, N\} \end{aligned} \quad (13)$$

For the column scrambling, a permutation function is defined as $\pi_c : \{1, 2, \dots, N\} \rightarrow \{1, 2, \dots, N\}$ that randomly shuffles the columns of $P_{shuffled_rows}(\pi_r(i), j)$. The new image $P_{shuffled}$ after both row and column shuffling is given by Equation 14.

$$P_{shuffled}(\pi_r(i), \pi_c(j)) = P_{enc}(i, j), \quad \forall i \in \{1, 2, \dots, M\}, \forall j \in \{1, 2, \dots, N\} \quad (14)$$

Pixel value permutation

It changes the order of pixel values in the image without changing their spatial locations. This technique disrupts the color distribution while maintaining the structural content. In this step, there are also two techniques of permutation such as flattening the image, and a permutation of pixel values is applied. First, flatten the image ($P_{shuffled}(\pi_r(i), \pi_c(j))$) into a one-dimensional vector $p = [p_1, p_2, \dots, p_{MN}]$, where $p_k = P_{enc}(i_k, j_k)$ for $k = 1, 2, \dots, MN$. Whereas, in the permutation of pixel values, first define a permutation function $\pi_v : \{1, 2, \dots, MN\} \rightarrow \{1, 2, \dots, MN\}$ that rearranges the pixel values. The new pixel vector $\mathbf{p}'$ after permutation is $\mathbf{p}'(\pi_v(k)) = p(k)$, $\forall k \in \{1, 2, \dots, MN\}$. After that, reshape the permuted vector $\mathbf{p}'$ back to the original image dimensions $M \times N$ to obtain the permuted image P_{perm} .

Rotation and flipping

To rotate an image $P_{shuffled}(\pi_r(i), \pi_c(j))$ by θ degrees around its center, each pixel $P_{enc}(i, j)$ is mapped to a new location (i', j') using the rotation matrix which is given in Equation 15.

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} i - i_c \\ j - j_c \end{bmatrix} + \begin{bmatrix} i_c \\ j_c \end{bmatrix} \quad (15)$$

where (i_c, j_c) is the center of the image, calculated as: $i_c = \frac{M+1}{2}$, $j_c = \frac{N+1}{2}$.

To flip the image, both horizontal and vertical flipping techniques are employed according to the Equations 16, and 17, respectively.

$$P_{flip_horizontal}(i, j) = P_{perm}(i, N - j + 1), \quad \forall i \in \{1, 2, \dots, M\}, \forall j \in \{1, 2, \dots, N\} \quad (16)$$

$$P_{flip_vertical}(i, j) = P_{flip_horizontal}(M - i + 1, j), \quad \forall i \in \{1, 2, \dots, M\}, \forall j \in \{1, 2, \dots, N\} \quad (17)$$

Algorithm 2 provides the pseudocode used to generate the confusion depicted in the image.

Diffusion phase

The objective of this phase is to disperse the influence of each pixel in I_s throughout the entire image for enhancing the overall security. In this research, this is achieved using a different complex mathematical operation. The detail of such operations is provided in the next subsections.

Affine transformation with nonlinear functions

Affine transformation is integrated with nonlinear functions to enhance diffusion. There are two major transformations in this process: (i) Affine transformation and (ii) nonlinear function application. Mathematical representations of these transformations are given in Equations 18, and 19, respectively.

$$A(i, j) = a \cdot \mathbf{P}_{flip_vertical}(i, j) + b \quad (18)$$

$$A(x, y) = \left[\left(a \cdot \mathbf{P}_{flip_vertical}(x, y) + b \right) \cdot \left[\sin \left(\frac{(x+y) \cdot \pi}{N \cdot M} \right) + \left(\frac{(x-y)^2}{\sqrt{N \cdot M}} \right)^p \right] \right] \quad (19)$$

Where a and b are constants, and p is a polynomial degree.

Discrete Cosine Transform (DCT) with complex modulation

Apply the 2D DCT to the $A(x, y)$ using Equation 20.

$$D(x, y) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} A(x, y) \cdot \cos \left[\frac{\pi(2i+1)}{2N} \right] \cdot \cos \left[\frac{\pi(2j+1)}{2M} \right] \quad (20)$$

After applying 2d DCT, use a complex modulation function to the DCT coefficients to get a second pre-ciphertext image ($\mathbf{D}'(x, y)$) using Equation 21.

$$\mathbf{D}'(x, y) = \mathbf{D}(x, y) \cdot \left(\exp \left[\frac{\pi}{N+M} \right] \cdot \sin \left(\frac{\pi^2}{N^2 \cdot M^2} \right) \right) \quad (21)$$

Now, apply the inverse 2D DCT using Equation 22 to obtain the third ciphertext image ($\mathbf{D}''(u, v)$).

$$\begin{aligned} \mathbf{D}''(x, y) = \frac{2}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \mathbf{D}'(x, y) \\ \cdot \cos \left[\frac{\pi(2x+1)}{2N} \right] \cdot \cos \left[\frac{\pi(2y+1)}{2M} \right] \end{aligned} \quad (22)$$

Wavelet transform with random phase modulation

In this research, the wavelet transform is used to decompose $\mathbf{D}''(x, y)$ into different frequency sub-bands (LL, LH, HL, and HH) which consist of both high- and low-frequency components. The 2D wavelet transform is implemented using Equation 23.

$$W(j, k, a, b) = \sum_{x, y} \mathbf{D}''(x, y) \cdot \psi_{j, k}(x - a, y - b) \quad (23)$$

Where $W(j, k, a, b)$, and $\psi_{j, k}(x - a, y - b)$ are the wavelet coefficients, and wavelet function, respectively. The wavelet function is shifted by (a, b) and scaled according to indices j and k . The random phase modulation is applied to the wavelet coefficients to for the modification of phase information of the image components using Equation 24.

$$\begin{aligned} \mathbf{W}'(j, k, a, b) = W(j, k, a, b) \cdot \left[\exp \left(i \cdot \frac{\pi \cdot (a + b)}{M \cdot N} \right) \right. \\ \left. \cdot \left(\sin \left(\frac{2\pi \cdot j \cdot k}{N} \right) + \frac{a}{b} \cdot \cos \left(\frac{\pi \cdot (a^2 + b^2)}{M \cdot N} \right) \right) \right] \end{aligned} \quad (24)$$

Where a and b are constants. The inverse wavelet transform is applied using Equation 25.

$$C(x, y) = \sum_{j, k} \sum_{a, b} \mathbf{W}'(j, k, a, b) \cdot \psi_{j, k}(x - a, y - b) \quad (25)$$

Bilinear transformation

The purpose of bilinear transformations is to create non-linear relationships between image pixels in a way that is not easily reversible without exact replica of the secret keys and the knowledge of the transformation parameters. Furthermore, adding a bilinear transformation to the encryption process enhances the security of the enciphered images. The bilinear transformation for $C(x, y)$ is given in Equation 26.

$$B(i, j) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} C(x, y) \cdot \left[\frac{(i+k) \cdot (j+l)}{N \cdot M} \right] \quad (26)$$

To increase further complexity, a complex weighting function is applied to the transformed pixel values using Equation 27.

$$\begin{aligned} \mathbf{C}'(i, j) = \left[B(i, j) \cdot \left[\frac{1}{1 + \exp \left(-\frac{(i+j)}{\sqrt{N \cdot M}} \right)} \right. \right. \\ \left. \left. + \cos \left(\frac{\pi \cdot (i^2 + j^2)}{N \cdot M} \right) \right] \right] \end{aligned} \quad (27)$$

Nonlinear polynomial mapping

In the last step of the proposed encryption framework, a nonlinear polynomial mapping is applied to change the pixel values based on higher-order polynomials. This introduces non-linearity and makes the transformation difficult to reverse without knowing the exact polynomial used. A nonlinear polynomial mapping is applied to each pixel value to produce the final ciphertext image ($C_f(I, j)$) using Equation 28.

$$C_f(i, j) = \left[\left(C(i, j)^3 + \frac{\pi \cdot (i \cdot j)}{N + M} \right) \cdot \left[\exp \left(\frac{i^2 + j^2}{N \cdot M} \right) + \sin \left(\frac{2\pi \cdot i \cdot j}{N \cdot M} \right) \right] \right] \quad (28)$$

The pseudocode of the proposed diffusion phase is given in Algorithm 3. The entire encryption framework provides a significantly robust non-linear relationship between the original, and the final ciphertext image. The generated ciphertext images are displayed in Figure 4, where all plaintext information is entirely concealed, and the original image is unrecognizable. This demonstrates the effectiveness of the proposed framework in encrypting the plaintext image by disrupting pixel correlations.

Experimental results and analysis

To test and analyze the proposed encryption framework for securing digital images, four test images are used: X-ray, Tumor, Ultrasound, and Electrocardiogram (ECG) signal images. The entire encryption framework is implemented in MATLAB 2018 on a system with the following hardware specifications: 8 GB RAM, 512 GB SSD, 11th generation Intel Core i5 processor at 2.4 GHz, running Windows 11. The effectiveness of the encryption framework is assessed through various statistical security analyses, including entropy, histogram, lossless analysis, and key sensitivity. The statistical values for the encrypted images are averaged across the R, G, and B components. The framework is also tested against multiple cyberattacks including brute force, noise, and cropping attacks to show its resilience to these threats.

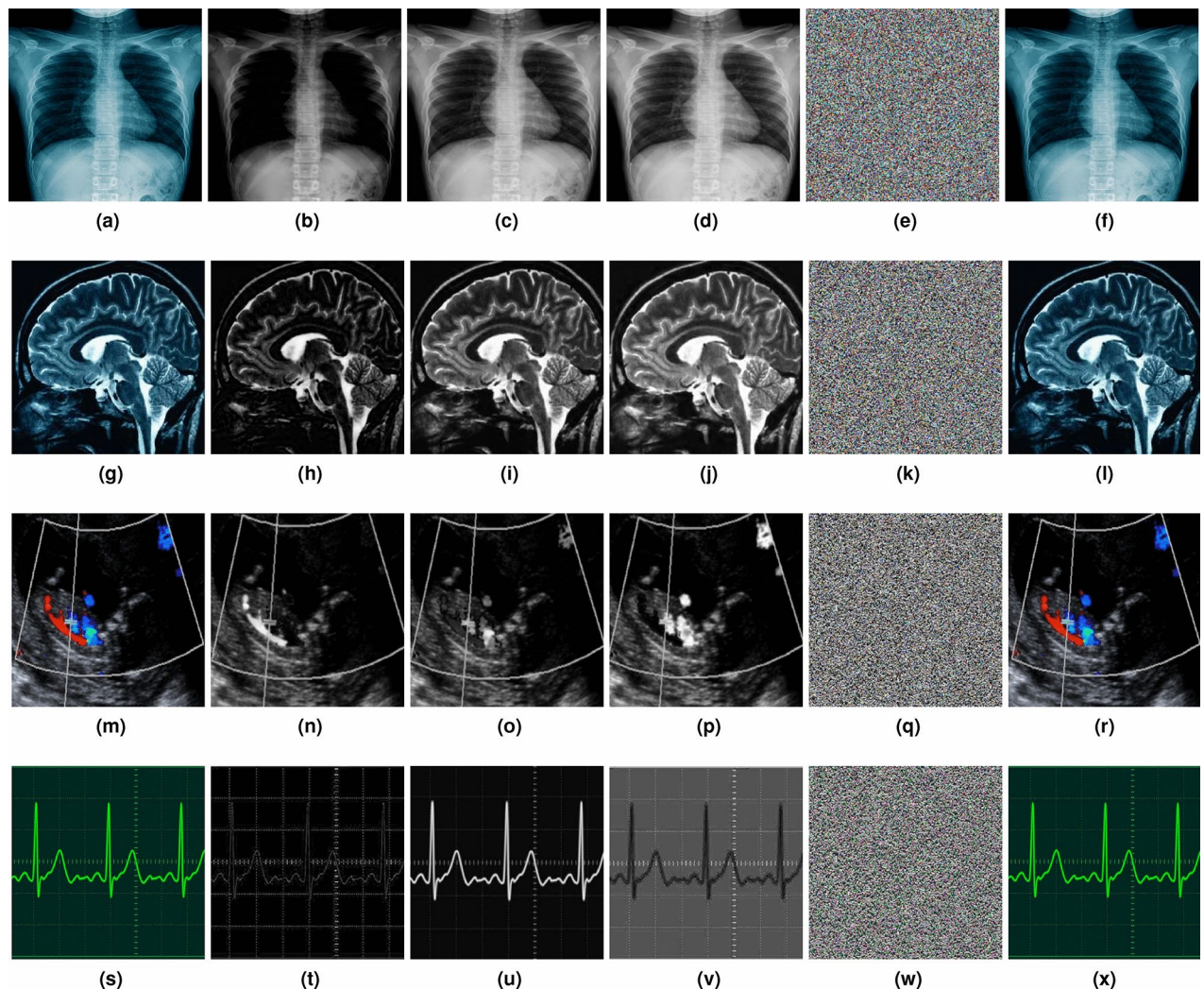


Fig. 4. (a,f,k,p) Plaintext images, (b-d, g-i, l-n, q-s) their corresponding R, G, B components, and (e, j, o, k) corresponding decrypted versions.

Algorithm test

The proposed encryption framework is tested on different plaintext color images including X-ray and tumor images with a resolution of 256×256 , as well as ultrasound and ECG signal images with a resolution of 512×512 having varying sizes such as 256×256 , and 512×512 . The quality of the encrypted images is then assessed visually. As shown in Figure 4, the plaintext information is completely concealed within the ciphertext images, with no visible patterns of the original data. This shows that the proposed encryption framework is effective in protecting plaintext information.

Entropy

Entropy measures the randomness in an image, with the ideal value varying based on the bit depth of the image. The maximum entropy value for an 8-bit image is 8, while for a binary (2-bit) image, it is 2. Mathematically, it can be calculated using Equation 29.

$$E_I = \sum_{L=0}^M p(i_L) \log_b p(i_L) \tag{29}$$

Where, E_I is the entropy of any image. $\log_b(p(i_L))$ is the logarithm of the probability $p(t_k)$ with a base b . The base b depends on the entropy calculation. For example, if $b = 2$, it is a binary logarithm, which is commonly used when dealing with bits. $\sum_{K=0}^N$ is the summation symbol which indicates that the sum is calculated over all possible pixel values L in the image ranges from 0 to M . For robust encryption, it is essential that the entropy value is as high as possible. A higher entropy value indicates greater randomness in an image, which contributes to stronger encryption. The relationship between high entropy and robust encryption is expressed in Equation 30.

$$\text{Entropy} \propto \text{Strong Encryption} \tag{30}$$

Table 3 presents entropy values for both the proposed and existing methods. The results show that the entropy values of the encrypted images from the proposed framework are closer to 8, indicating a higher level of randomness in the ciphertext compared to existing methods.

In addition to the global entropy calculation, the encrypted and plaintext images are divided into blocks of various sizes containing 16, 64, 256, and 1024 pixels in each block. The 3D entropy distribution plots for these blocks are displayed in Figure 5. Even with a block size of 4×4 , the entropy remains above approximately 3.58, indicating that no pattern of even 16 pixels repeats in the encrypted image, with each pixel appearing random. This results in a high global entropy, as shown in Table 3. In contrast, the 3D block-wise entropy distribution for the plaintext image in Figure 5(a-d) shows entropy dropping to zero for 4×4 and 8×8 blocks, revealing smooth patterns in the plaintext image. Furthermore, the global entropy for 100 plaintext and corresponding ciphertext images is displayed in the polynomial surface fit plots in Figure 5(i, j). It can be observed that the global entropy of the plaintext images falls within the range of approximately [6.7981, 6.79983], while for the corresponding ciphertext images, it ranges from roughly [7.9994, 7.9995].

Correlation analysis

Correlation evaluates how adjacent pixels relate to each other in different orientations. In general, plaintext images exhibit high pixel correlation because of the similarity between neighboring pixels. However, in image encryption, it is crucial to break this pixel correlation to protect the plaintext information securely. Reduced correlation indicates a more robust encryption, as it signifies that the encryption process has effectively broken the inherent relationships between pixels, enhancing the security of the image. The pixel correlation coefficients for horizontally (H), vertically (V), and diagonally (D) adjacent pixels are calculated using Equations 31, 32, and 33 to assess the effectiveness of the proposed encryption framework.

$$\text{Corr}_{\text{horizontal}} = \frac{\sum_{j=1}^{W-1} (I(i, j) - \bar{I}_i)(I(i, j + 1) - \bar{I}_i)}{(W - 1)\sigma_i^2} \tag{31}$$

$$\text{Corr}_{\text{vertical}} = \frac{\sum_{i=1}^{H-1} (I(i, j) - \bar{I}_j)(I(i + 1, j) - \bar{I}_j)}{(H - 1)\sigma_j^2} \tag{32}$$

Image size	Images	32	28	44	45	46	Proposed
$256 \times 256 \times 3$	Xray	7.9967	7.9919	7.9936	7.9955	7.9949	7.9993
	Tumor	7.9946	7.9966	7.9919	7.9977	7.9991	7.9994
$512 \times 512 \times 3$	Ultrasound	7.9961	7.9979	7.9987	7.9949	7.9966	7.9992
	ECG signal	7.9916	7.9910	7.9933	7.9969	7.9982	7.9991

Table 3. Entropy analysis.

```

1: Input: Scrambled image  $I_s$  ( $M \times N$ )
2: Define constants  $a, b$ , polynomial degree  $p$ 
3: Affine Transformation:
4: for  $i = 1$  to  $M$  do
5:   for  $j = 1$  to  $N$  do
6:      $A(i, j) \leftarrow aI_s(i, j) + b \leftarrow$  Equation 18
7:   end for
8: end for
9: Nonlinear Function:
10: for  $x = 1$  to  $M$ ,  $y = 1$  to  $N$  do
11:    $A(x, y) \leftarrow$  Equation 19
12: end for
13: DCT and Complex Modulation:
14: for  $u = 0$  to  $N - 1$ ,  $v = 0$  to  $M - 1$  do
15:    $D(u, v) \leftarrow$  Equation 20
16:
17:    $D'(u, v) \leftarrow$  Equation 21
18: end for
19: Inverse DCT:
20: for  $x = 0$  to  $N - 1$ ,  $y = 0$  to  $M - 1$  do
21:    $D''(x, y) \leftarrow$  Equation 22
22: end for
23: Wavelet Transform and Complex Weighting:
24: for  $j = 1$  to  $J$ ,  $k = 1$  to  $K$ ,  $a = 1$  to  $M$ ,  $b = 1$  to  $N$  do
25:    $W(j, k, a, b) \leftarrow$  Equation 23
26:
27:    $W'(j, k, a, b) \leftarrow$  Equation 24
28: end for
29: Inverse Wavelet Transform:
30: for  $x = 1$  to  $M$ ,  $y = 1$  to  $N$  do
31:    $C(x, y) \leftarrow$  Equation 25
32: end for
33: Bilinear Transformation:
34: for  $i = 1$  to  $M$ ,  $j = 1$  to  $N$  do
35:    $B(i, j) \leftarrow$  Equation 26
36: end for
37: Complex Weighting and Nonlinear Polynomial Mapping:
38: for  $i = 1$  to  $M$ ,  $j = 1$  to  $N$  do
39:    $C'(i, j) \leftarrow$  Equation 27
40:
41:    $C_f(i, j) \leftarrow$  Equation 28
42: end for
43: Output: Final ciphertext image  $C_f$ 

```

Algorithm 3. Diffusion phase.

$$\text{Corr}_{\text{diagonal}} = \frac{\sum_{i=1}^{H-1} \sum_{j=1}^{W-1} (I(i, j) - \bar{I})(I(i+1, j+1) - \bar{I})}{(N-1)\sigma^2} \quad (33)$$

In Equations 31–33, $I(i, j)$ represents the pixel value at row i and column j . W and H denote the width and height of the image, respectively; $\bar{I}_i$ is the mean pixel value for row i or column j ; and σ_i^2 is the variance of pixel values in row i . Table 4 displays various correlation values, showing that the ciphertext images exhibit minimal correlations in all directions compared to existing encryption methods, with coefficients approaching zero. This indicates that the proposed encryption method is more effective at randomizing pixel values and disrupting the patterns present in the plaintext information.

A visual analysis using scatter plots shows the statistical correlation analysis. Figure 6 compares the pixel correlations of the plaintext and ciphertext images. The closely clustered points in Figures (a–c, g–i) indicate a high correlation among plaintext pixels, while the more scattered points in Figures (d–f, j–l) demonstrate very little correlation in the ciphertext pixels.

Key sensitivity analysis

For any encryption framework, it is crucial that the secret keys used are highly sensitive. Sensitivity to secret keys means that even a tiny change in the keys makes it impossible to successfully decrypt the plaintext information. In this work, a total of fifteen secret keys ($\varpi_1, \varpi_2, \dots, \varpi_6, \sigma_1, \rho_1, \beta_1, \sigma_2, \rho_2, \beta_2, \Xi, \xi, u, k, p$) are used. To perform the key sensitivity analysis, a minute change of $\Delta = 10^{-15}$

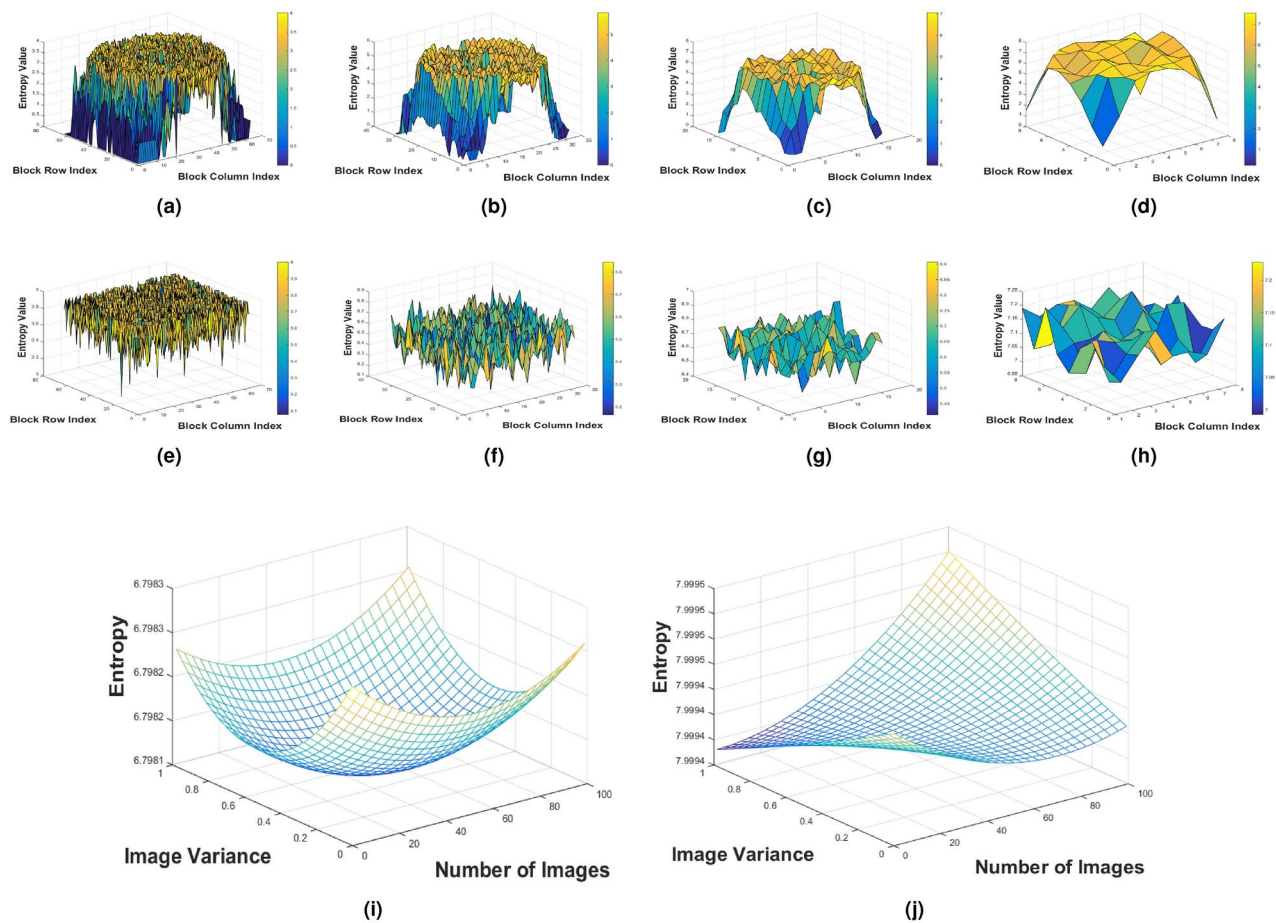


Fig. 5. Polynomial surface fit plots for entropy values of 100 original and corresponding encrypted images.

Image size	Images	Directions	32	28	44	45	46	Proposed
256 × 256 × 3	Xray	H	0.0023	0.0015	0.0014	0.0016	0.0028	0.0001
		V	0.0016	0.0019	0.0022	0.0019	0.0029	0.0002
		D	0.0011	0.0031	0.0026	−0.0016	0.0019	−0.0003
	Tumor	H	0.0014	0.0013	−0.0023	−0.0015	0.0014	−0.0013
		V	0.0011	0.0021	−0.0016	0.0031	−0.0020	0.0001
		D	0.0011	0.0013	−0.0011	0.0018	−0.0011	−0.0007
512 × 512 × 3	Ultrasound	H	0.0021	−0.0022	−0.0013	0.0022	−0.0016	−0.0002
		V	0.0011	0.0019	−0.0017	0.0031	0.0010	0.0003
		D	0.0016	0.0021	−0.0033	0.0017	0.0033	0.0006
	ECG signal	H	0.0023	−0.0014	−0.0013	−0.0017	0.0032	−0.0013
		V	0.0016	−0.0011	0.0031	0.0008	0.0031	0.0005
		D	0.0034	0.0064	0.0046	0.0044	−0.001	−0.0003

Table 4. Correlation analysis.

is introduced to each secret key. The modified keys are $\varpi'_1 = \varpi_1 + \Delta, \varpi'_2 = \varpi_2 + \Delta, \dots, \varpi'_6 = \varpi_6 + \Delta, \sigma'_1 = \sigma_1 + \Delta, \rho'_1 = \rho_1 + \Delta, \beta'_1 = \beta_1 + \Delta, \sigma'_2 = \sigma_2 + \Delta, \rho'_2 = \rho_2 + \Delta, \beta'_2 = \beta_2 + \Delta, \Xi' = \Xi + \Delta, \xi' = \xi + \Delta, u' = u + \Delta, k' = k + \Delta, p' = p + \Delta$. These modified keys are then used to decrypt the plaintext image from the ciphertext image.

The resulting decrypted images, which are recovered using the modified secret keys, are shown in Figure 7. It can be observed in Figure 7(c) that the decrypted images contain no recognizable information from the original plaintext image, which indicates that even a tiny modification to the secret keys renders decryption completely ineffective. This demonstrates that the secret keys used in the proposed encryption framework are

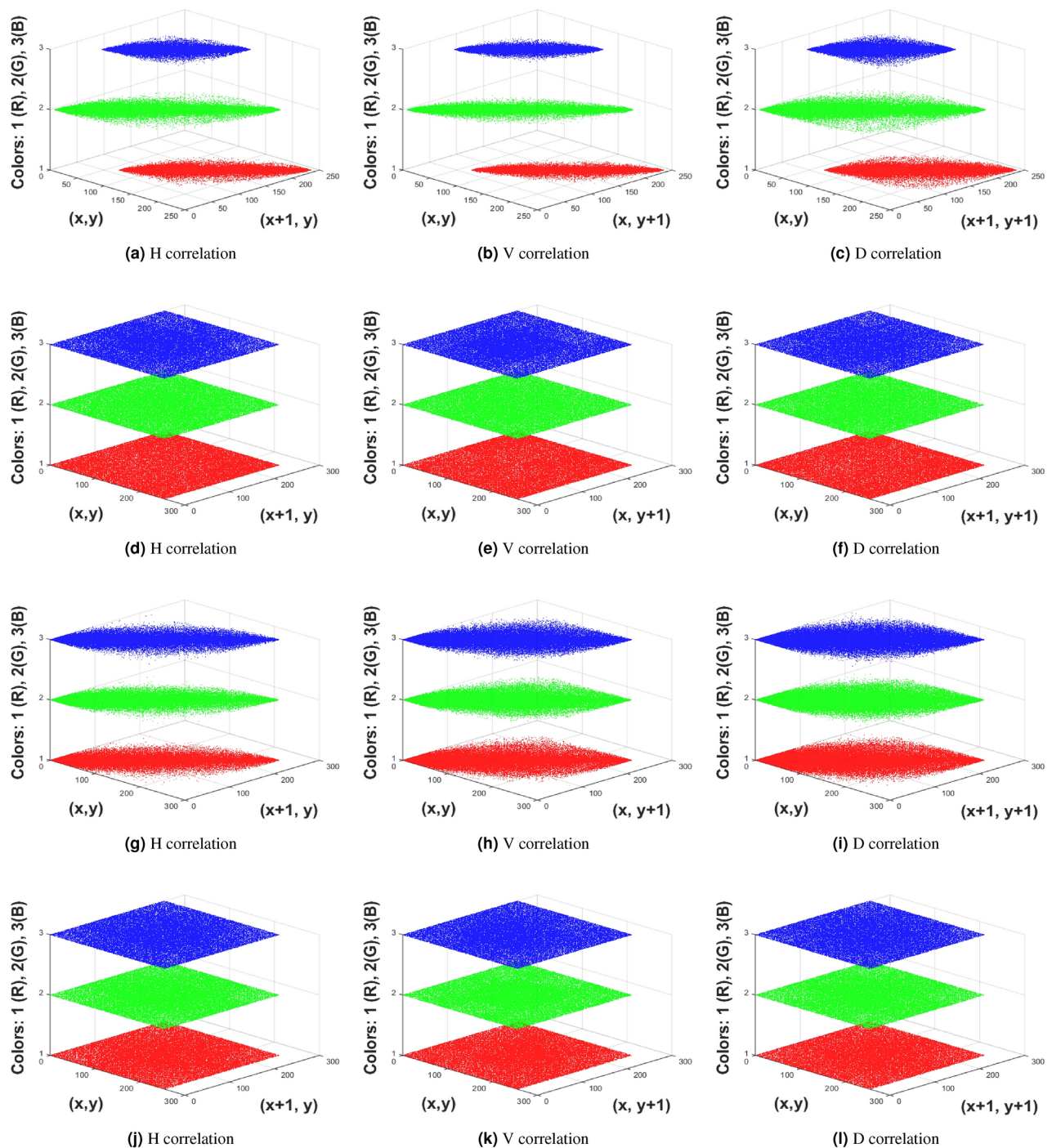


Fig. 6. Correlation in multi directions (a–c) Correlation in Tumor image, (d–f) Correlation in encrypted Tumor image, (g–i) (a–c) Correlation in Xray image, (j–l) Correlation in encrypted Xray image.

highly sensitive. To successfully decrypt the plaintext information, the exact keys must be used. Figure 7 (d) shows the decrypted image using the correct keys, where it is evident that the image is successfully recovered.

Key space analysis

Keyspace analysis refers to the size of the secret keys used in an encryption framework and is also related to brute force attacks. In a brute force attack, an adversary attempts every possible combination of secret keys to breach the encryption. To determine the size of the secret keys in the proposed encryption framework, first assess key sensitivity as explained in Section 5.4. In this proposed framework, the sensitivity of each key is at least 10^{-15} , meaning the size of each secret key is 10^{+15} . Therefore, the total key space for the fifteen keys is $10^{15 \times 15}$, which

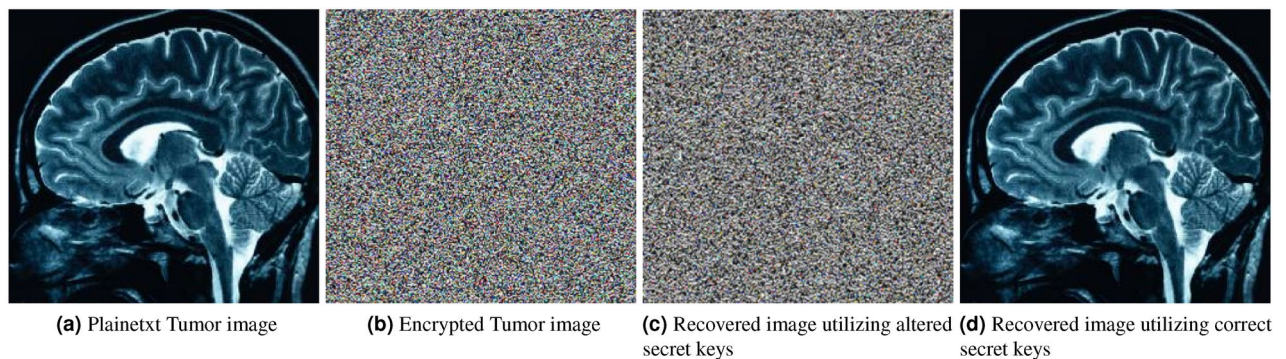


Fig. 7. Key sensitivity analysis.

is approximately $2^{747.43}$. According to Alvarez's⁴⁷ criteria for key space, any encryption algorithm with a key space equal to or greater than 2^{100} is considered secure against brute force attacks.

Noise and cropping attack analysis

To make decryption fail, attackers often attempt to modify the ciphertext image by adding external noise to it or cropping a portion of the ciphertext image.

In the ciphertext image, 10% salt and 10% pepper noise are added randomly to alter each pixel's value. To evaluate the proposed encryption framework, first, random noise is added to the ciphertext image pixels. For each pixel at position (i, j) in the image I , the noisy image I' will be calculated using Equation 34.

$$I'(i, j) = \begin{cases} 0 & \text{with probability 0.1 (pepper noise)} \\ 255 & \text{with probability 0.1 (salt noise)} \\ I(i, j) & \text{with probability } 1 - 0.2 = 0.8 \text{ (no change)} \end{cases} \quad (34)$$

After adding noise to the ciphertext image pixels, the plaintext image is recovered from the noisy version of the ciphertext. The visualized results are displayed in Figure 8. The decrypted image, shown in Figure 8(e), demonstrates that the information in the plaintext image is still visible, although some noise remains in the recovered image. This indicates that while the exact pixel values are not fully restored, the content of the plaintext image is still easily recognizable. This result shows that the proposed encryption framework is effective in resisting noise attacks.

To perform a cropping attack analysis on the encryption framework, 20% portion of the ciphertext image is cropped and then attempt to decrypt the plaintext image from this cropped version. The decrypted image is shown in Figure 8(f). Based on your cropping attack analysis, the results indicate that the proposed encryption framework is effective at resisting cropping attacks. Despite removing 20% of the ciphertext image, the decryption process is able to produce an output where the plaintext image remains largely recognizable. This suggests that the encryption framework is resilient to partial data loss, as significant portions of the original content are still recoverable from the cropped ciphertext. The ability of the framework to maintain the integrity of the plaintext image under such conditions shows its robustness and effectiveness in handling attacks that involve manipulating or removing portions of the ciphertext.

Several noise and cropping curves are generated for different images to demonstrate how much plaintext data can be recovered after adding external salt and pepper noise at varying percentages from 10% to 50%, with a step size of 2%. Likewise, for the cropping attack analysis, the image is cropped from 10% to 50% in increments of 2%. Figure 9 illustrates that even with the addition of 50% salt and pepper noise to the ciphertext, the proposed framework is capable of decrypting more than 90% of the plaintext information. Similarly, after cropping 50% of the ciphertext image, over 90% of the plaintext information can still be recovered.

Histogram analysis

Histogram analysis is a technique used to assess the security and effectiveness of an encryption algorithm by analyzing the pixel distribution in the image⁴⁸. For an unencrypted image, the histogram shows different patterns corresponding to the visual content of the image. For an encrypted image, the histogram should be uniform and distinctly different from that of the plaintext image. This uniformity indicates that the encryption algorithm has effectively encrypted the original pixel values, which makes it difficult for an eavesdropper to extract any meaningful information.

In the proposed research, several histograms are generated for the different color components of the images. Figure 10 displays the histograms of the unencrypted color components, while Figure 11 shows those of the encrypted color components. The histograms of the encrypted color components show a uniform pixel distribution and differ significantly from those of the plaintext color components, indicating that the proposed encryption framework effectively resists histogram attacks.

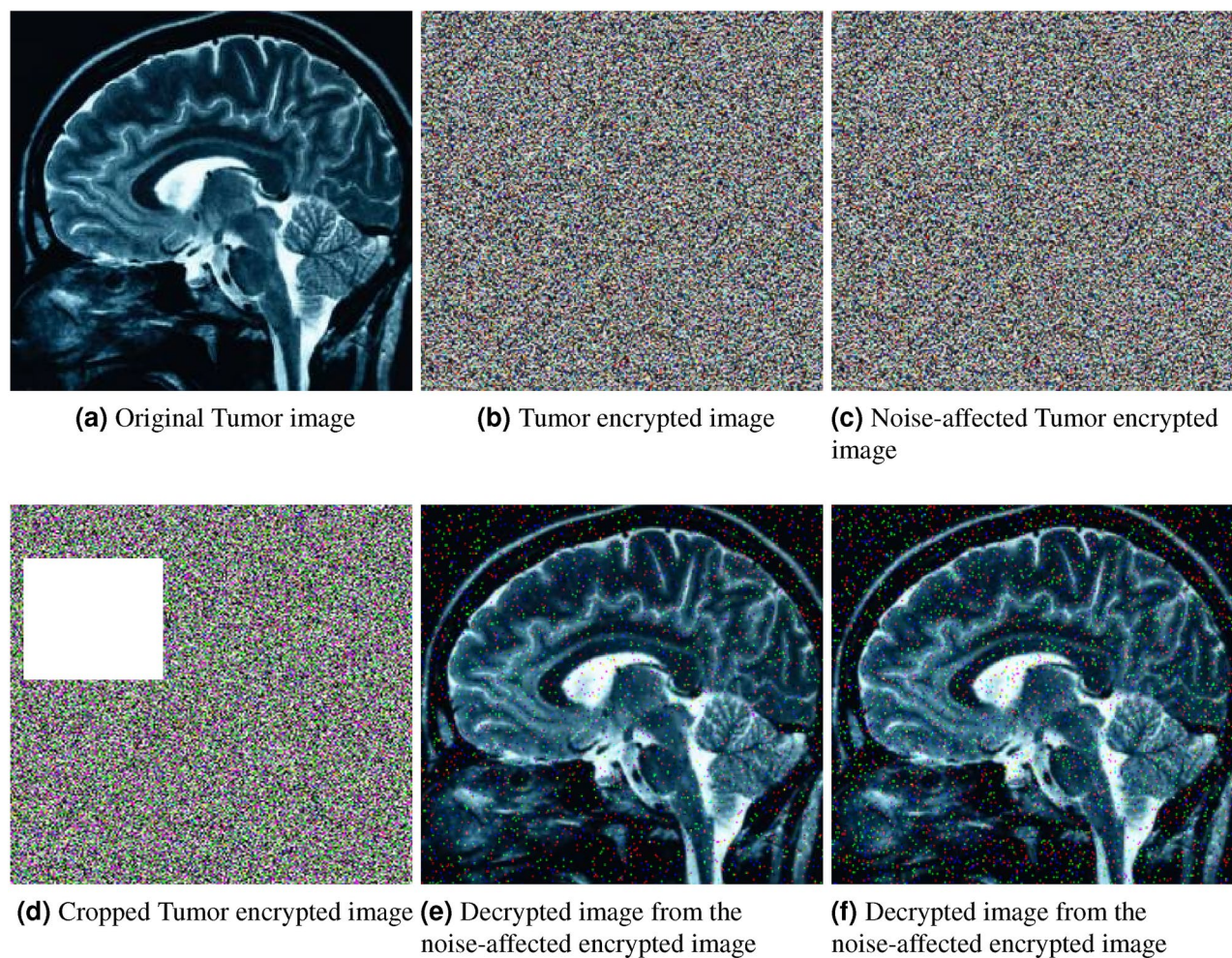


Fig. 8. Noise and cropping attack analysis.

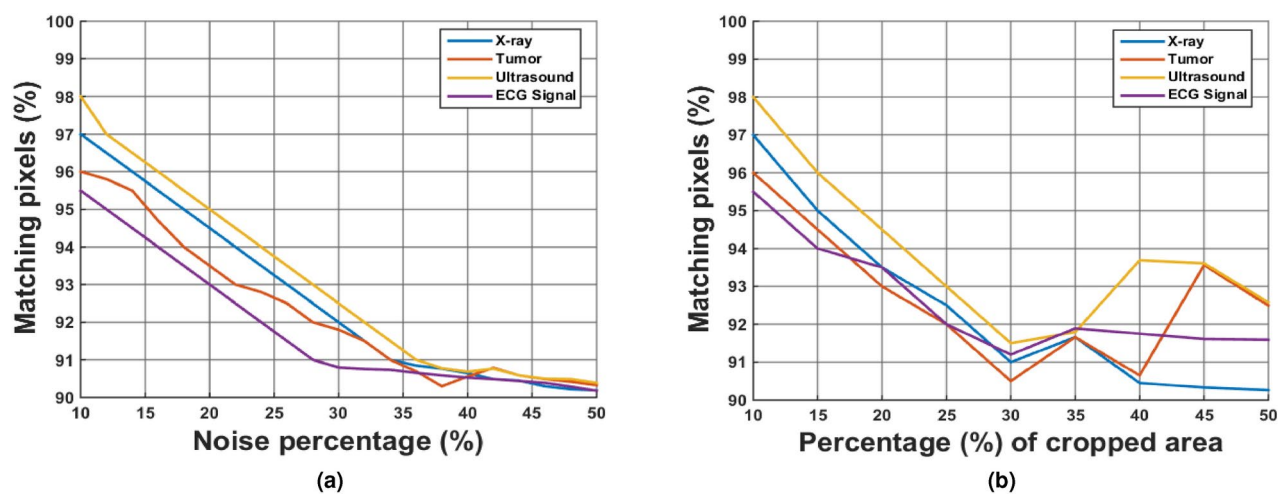


Fig. 9. Noise and cropping attack analysis for different medical images.

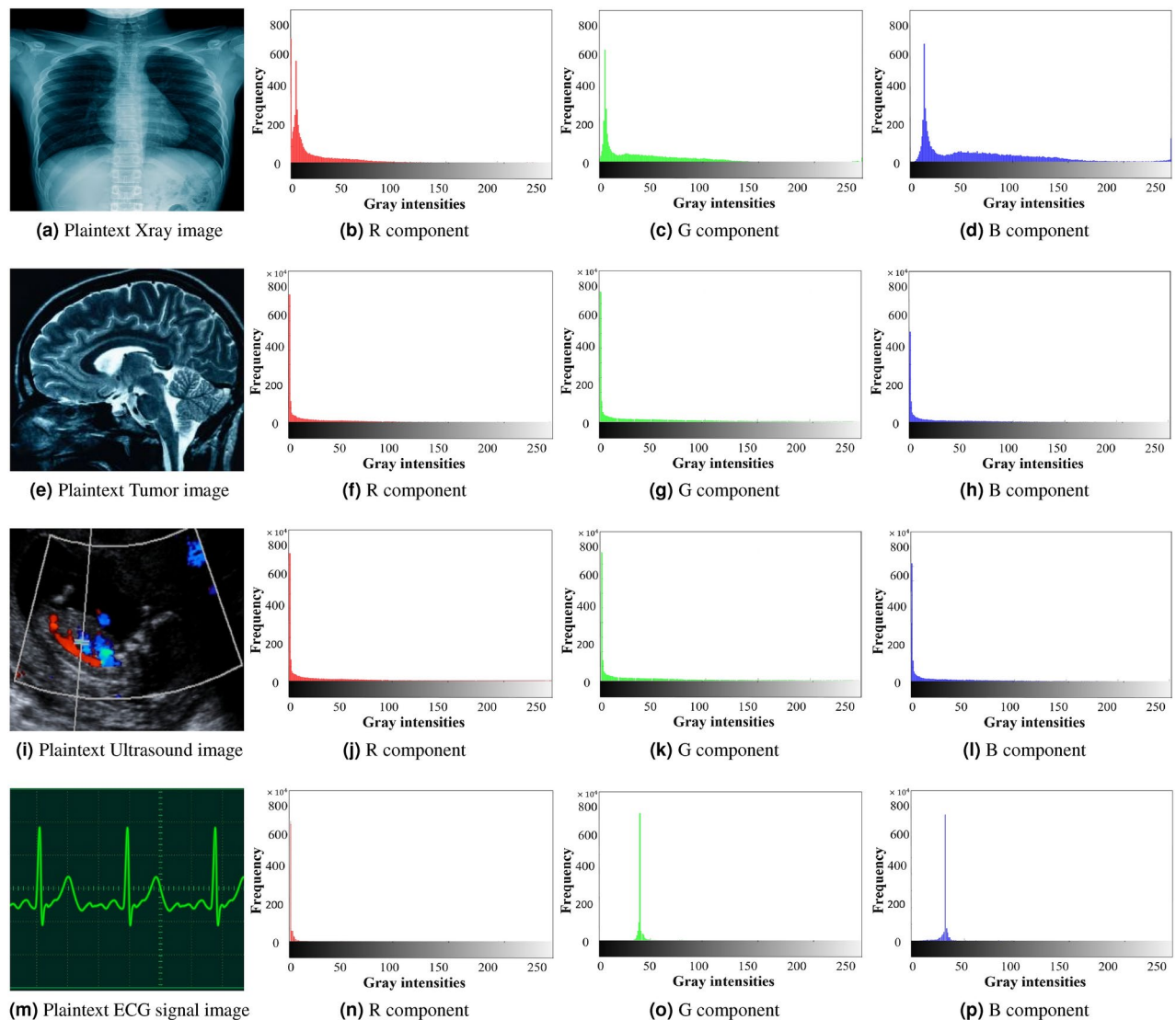


Fig. 10. Histograms of the plaintext color images.

Histogram variance analysis

Histogram variance analysis is used to statistically evaluate the effectiveness of an encryption algorithm by examining the distribution of pixel intensities in the encrypted image. Mathematically, histogram variance can be calculated using Equation 35.

$$\sigma^2 = \sum_{i=0}^{L-1} p(i) \cdot (i - \mu)^2 \quad (35)$$

Where, σ^2 is the histogram variance, L is the number of possible intensity levels, $p(i)$ is the probability of occurrence of intensity level i , and μ is the mean intensity of the histogram which can be calculated as $\mu = \sum_{i=0}^{L-1} i \cdot p(i)$. The histogram variance σ^2 quantifies how pixel intensities deviate from the mean intensity value. For a strong encryption, the histogram variance should be low which indicates a uniform distribution of pixel values. Table 5 shows that the proposed encryption framework achieves lower histogram variance values than existing schemes, indicating its superior performance.

Lossless analysis

Lossless analysis determines whether any plaintext information is lost after decrypting the plaintext image from the ciphertext image. To assess whether the proposed encryption method is lossless or lossy, two well-known metrics are used: Peak Signal-to-Noise ratio (PSNR) and Mean Squared Error (MSE). PSNR measures the

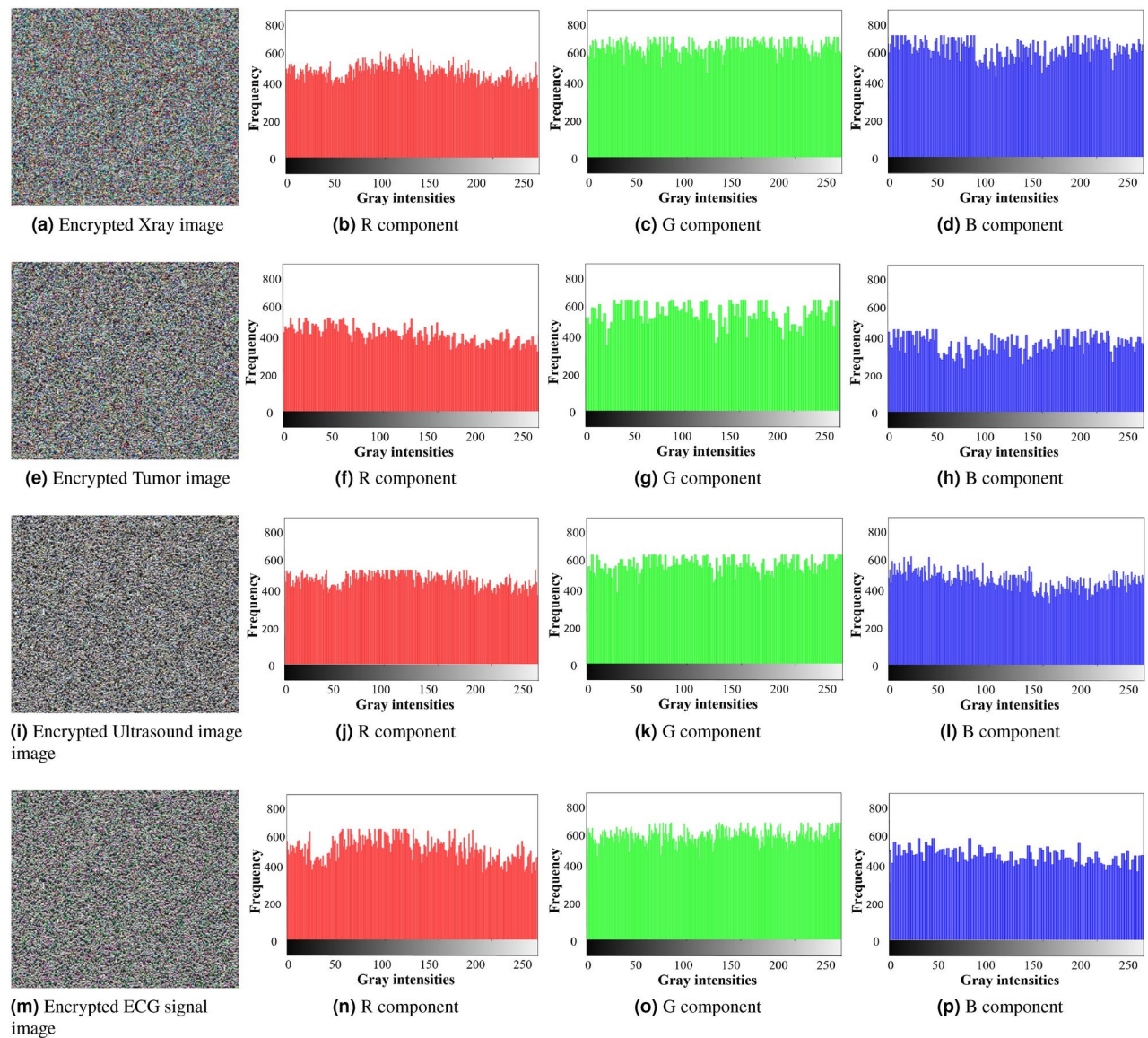


Fig. 11. Histogram of the encrypted color components.

Image size	Images	32	28	44	45	46	Proposed
$256 \times 256 \times 3$	Xray	260.36	261.46	261.30	260.15	259.16	258.31
	Tumor	261.03	262.34	265.66	270.16	266.03	257.01
$512 \times 512 \times 3$	Ultrasound	266.31	261.09	270.16	267.78	260.33	259.46
	ECG signal	260.19	265.19	270.19	259.66	260.11	259.88

Table 5. variance analysis.

similarity between the plaintext image and the decrypted image, while MSE calculates the difference between these images. Mathematically, PSNR and MSE can be computed using Equations 36 and 37, respectively.

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) \quad (36)$$

$$\text{MSE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(i, j) - K(i, j))^2 \quad (37)$$

PSNR							
Image size	Images	32	28	44	45	46	Proposed
256 × 256 × 3	Xray	50.5	59.12	63.32	71.65	50.3	∞
	Tumor	55.61	66.15	70.64	84.64	59.16	∞
512 × 512 × 3	Ultrasound	59.61	61.76	68.97	74.68	68.31	∞
	ECG signal	60.19	66.97	70.64	73.15	69.11	∞
MSE							
Image size	Images	32	28	44	45	46	Proposed
256 × 256 × 3	Xray	10.69	15.19	16.59	5.16	21.34	0
	Tumor	11.65	16.19	18.33	3.59	22.66	0
512 × 512 × 3	Ultrasound	16.11	19.49	20.97	1.69	24.09	0
	ECG signal	19.19	18.16	20.19	4.98	24.67	0

Table 6. Lossless analysis.

Image size	Images	32	28	44	45	46	Proposed
256 × 256 × 3	Xray	0.014	0.220	0.362	0.0001	0.79	0.002
	Tumor	0.016	0.259	0.038	0.0002	0.89	0.003
512 × 512 × 3	Ultrasound	0.030	0.430	0.636	0.002	1.68	0.003
	ECG signal	0.035	0.510	0.729	0.004	1.59	0.005

Table 7. Computational time analysis.

Where, $I(i, j)$ and $K(i, j)$ represent the pixel values at position (i, j) in the plaintext and decrypted images, respectively, and MAX is the maximum possible pixel value of the image (e.g., 255 for an 8-bit grayscale image). Table 6 presents the PSNR and MSE values for both the proposed and existing encryption schemes. The MSE values for the encryption framework proposed in⁴⁴ are nearly zero for each corresponding plaintext color image. In contrast, for the proposed encryption framework, all PSNR values are infinite and all MSE values are zero, indicating no difference between the decrypted image and the original plaintext image. This demonstrates that the proposed encryption algorithm is lossless.

Computational time analysis

Apart from statistical security analysis and conducting cyberattacks on the ciphertext images to assess the effectiveness of the proposed encryption framework, it is also essential to evaluate the encryption scheme’s computational complexity. For real-time applications and low-memory devices, such as IoT devices, the encryption framework must minimize the time required to encrypt plaintext images. To determine the computational complexity of the proposed encryption framework, MATLAB commands ‘tic’ and ‘toc’ are used to measure the elapsed time between two points in a script.

The ‘tic’ command records the current time when it is executed, capturing the precise time in seconds since a fixed reference point in the past. In contrast, the ‘toc’ command calculates the time elapsed since the most recent ‘tic’ command was executed. Mathematically, if T_{start} and T_{end} represent the times at which ‘tic’ and ‘toc’ are executed, respectively, then the elapsed time ΔT is given by Equation 38.

$$\Delta T = T_{end} - T_{start} \tag{38}$$

Table 7 presents the computational complexity analysis for the proposed and existing encryption algorithms. It can be observed that the encryption algorithm proposed in⁴⁵ performs slightly better than the proposed encryption framework. However, the proposed encryption framework is still suitable for real-time and IoT applications that require fast processing, as it can encrypt plaintext images of sizes 256 × 256 and 512 × 512 in under one second. Moreover, apart from the encryption scheme in⁴⁵, the proposed encryption framework outperforms other comparable encryption schemes in terms of computational complexity.

Discussion

The proposed work consists of two main components: (i) securing secret keys from potential attackers using Quantum Key Distribution (QKD), and (ii) protecting plaintext medical images through the integration of multiple encryption techniques. The QKD is utilized alongside One-Time Pad (OTP) encryption to protect the secret keys, and the proposed research mathematically demonstrates that these keys are fully secured against

cyberattacks. Additionally, the integration of multiple encryption techniques aims to maximize randomness in the plaintext data to enhance protection against cyber threats.

For instance, using a single encryption method, such as a permutation operation that makes the encryption vulnerable to various attacks. When only a permutation function is applied to a plaintext image, the process remains insecure, as it only rearranges pixel positions without altering their values. This means that key patterns and structural information within the original image are largely preserved which makes it easier for attackers to reconstruct the image. Moreover, a ciphertext-only attack (COA) becomes possible when only a permutation is employed. In such cases, an attacker can take advantage of the statistical characteristics of the image, including patterns in color distribution or variations in intensity, to partially or completely reconstruct the original image. One more significant weakness of a permutation-only encryption method is its vulnerability to known-plaintext attacks (KPA). When an attacker has access to both the encrypted image and its corresponding plaintext image, they can analyze the pixel rearrangements to determine the permutation pattern. This ability to exploit statistical features makes the encryption scheme highly susceptible to reconstruction efforts. Furthermore, histogram analysis also presents a significant threat to permutation-only encryption. Since permutation does not change pixel values, the histogram of the encrypted image remains identical to that of the original image. This similarity enables attackers to analyze the histogram which provides statistical clues about the structure and content of the original image. Finally, the lack of diffusion in permutation-only encryption schemes makes them particularly weak. According to Claude Shannon⁴⁹, an encryption scheme can be considered secure only if it incorporates both confusion and diffusion operations. Therefore, the proposed encryption framework effectively integrates multiple encryption methods to create both confusion and diffusion in the plaintext data. This ensures strong security with low latency, as demonstrated by statistical security analyses, computational evaluations, and comparisons of the proposed encryption scheme's strength with existing encryption methods presented in Section 5.

Conclusion

This research presented a new hybrid encryption framework that combines quantum and classical cryptographic techniques for the secure transmission of medical images in IoT-based telemedicine networks. By leveraging QKD for the generation of secure shared keys and integrating classical cryptographic methods, the proposed framework addressed several security concerns such as weak key management, and weak security of the digital data. The proposed framework employed multiple encryption techniques, such as pixel shuffling, bit-plane extraction, and various transformation techniques for the enhancement of the confusion and diffusion between the plaintext image pixels. Experimental results demonstrated the robustness of the proposed encryption framework. The proposed encryption framework achieved impressive values in entropy, correlation, and key space, which are 7.999, 0.0001, and $2^{741.67}$, respectively. Moreover, the proposed framework is also tested against different cyber threats, including noise, cropping, and brute-force attack. It is found that after the addition of the external noise and cropping various portions of the ciphertext image, the proposed framework remains capable of decrypting over 90% of the plaintext image. This also showed its capability to protect sensitive medical data against unauthorized access and tampering. Additionally, the proposed encryption framework is well-suited for real-time and IoT applications where fast processing speed is required. This is demonstrated by the computational complexity analysis, where it is observed that the proposed encryption framework can encrypt plaintext color images of sizes 256×256 or 512×512 within one second.

The proposed encryption framework can be further enhanced by incorporating visual cryptography using DWT. Additionally, to further reduce computational time, Artificial Intelligence (AI) will be integrated to identify and focus on the Region of Interest (ROI) for encryption.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 3 October 2024; Accepted: 3 December 2024

Published online: 28 December 2024

References

- Kim, S., Choi, B. & Lew, Y. K. Where is the age of digitalization heading? the meaning, characteristics, and implications of contemporary digital transformation. *Sustainability* **13**, 8909 (2021).
- Liu, F. et al. Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications. *IEEE Wireless communications* **20**, 14–22 (2013).
- Shafique, A., Mehmood, A. & Elhadef, M. Survey of security protocols and vulnerabilities in unmanned aerial vehicles. *IEEE Access* **9**, 46927–46948 (2021).
- Douglas, M., Bailey, K., Leeney, M. & Curran, K. An overview of steganography techniques applied to the protection of biometric data. *Multimedia Tools and Applications* **77**, 17333–17373 (2018).
- Kaur, M., Singh, D., Kumar, V., Gupta, B. B. & Abd El-Latif, A. A. Secure and energy efficient-based e-health care framework for green internet of things. *IEEE Transactions on Green Communications and Networking* **5**, 1223–1231 (2021).
- Shafique, A. & Ahmed, F. Image encryption using dynamic s-box substitution in the wavelet domain. *Wireless Personal Communications* **115**, 2243–2268 (2020).
- Brahim, A. H., Pacha, A. A. & Said, N. H. Image encryption based on compressive sensing and chaos systems. *Optics & Laser Technology* **132**, 106489 (2020).
- Wang, J., Chen, J., Wang, F. & Ni, R. Optical image encryption scheme based on quantum s-box and meaningful ciphertext generation algorithm. *Optics Communications* **525**, 128834 (2022).
- Cherbal, S., Zier, A., Hebal, S., Louail, L. & Annane, B. Security in internet of things: a review on approaches based on blockchain, machine learning, cryptography, and quantum computing. *The Journal of Supercomputing* **80**, 3738–3816 (2024).

10. Shafique, A. A noise-tolerant cryptosystem based on the decomposition of bit-planes and the analysis of chaotic gauss iterated map. *Neural Computing and Applications* **34**, 16805–16828 (2022).
11. Mehmood, A., Shafique, A., Alawida, M. & Khan, A. N. Advances and vulnerabilities in modern cryptographic techniques: A comprehensive survey on cybersecurity in the domain of machine/deep learning and quantum techniques. *IEEE Access* **12**, 27530–27555 (2024).
12. Sonko, S., Ibekwe, K. I., Ilojany, V. I., Etukudoh, E. A. & Fabuyide, A. Quantum cryptography and us digital security: a comprehensive review: investigating the potential of quantum technologies in creating unbreakable encryption and their future in national security. *Computer Science & IT Research Journal* **5**, 390–414 (2024).
13. Mehmood, A. et al. A time-efficient and noise-resistant cryptosystem based on discrete wavelet transform and chaos theory: An application in image encryption. *Journal of Information Security and Applications* **78**, 103590 (2023).
14. Chamola, V., Jolfaei, A., Chanana, V., Parashari, P. & Hassija, V. Information security in the post quantum era for 5g and beyond networks: Threats to existing cryptography, and post-quantum cryptography. *Computer Communications* **176**, 99–118 (2021).
15. Abd El-Latif, A. A., Abd-El-Atty, B., Mazurczyk, W., Fung, C. & Venegas-Andraca, S. E. Secure data encryption based on quantum walks for 5g internet of things scenario. *IEEE Transactions on Network and Service Management* **17**, 118–131 (2020).
16. Rijmen, V. & Daemen, J. Advanced encryption standard. *Proceedings of federal information processing standards publications, national institute of standards and technology* **19**, 22 (2001).
17. Yu, S. et al. Development of modified blum-blum-shub pseudorandom sequence generator and its use in education. *Measurement Science Review* **22**, 143–151 (2022).
18. Zhou, S. et al. A novel image encryption cryptosystem based on true random numbers and chaotic systems. *Multimedia Systems* **1–18** (2022).
19. Muhammed, R. K. et al. Comparative analysis of aes, blowfish, twofish, salsa20, and chacha20 for image encryption. *Kurdistan Journal of Applied Research* **9**, 52–65 (2024).
20. Yang, C., Taralova, I., El Assad, S. & Loiseau, J.-J. Image encryption based on fractional chaotic pseudo-random number generator and dna encryption method. *Nonlinear Dynamics* **109**, 2103–2127 (2022).
21. Zia, U. et al. Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains. *International Journal of Information Security* **21**, 917–935 (2022).
22. Shafique, A. & Shahid, J. Novel image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps. *The European Physical Journal Plus* **133**, 331 (2018).
23. Hu, M., Li, J. & Di, X. Quantum image encryption scheme based on 2d sine-1 logistic chaotic map. *Nonlinear Dynamics* **111**, 2815–2839 (2023).
24. Zhang, W. et al. An image encryption scheme using self-adaptive selective permutation and inter-intra-block feedback diffusion. *Signal Processing* **151**, 130–143 (2018).
25. He, J., Zhu, H. & Zhou, X. Quantum image encryption algorithm via optimized quantum circuit and parity bit-plane permutation. *Journal of Information Security and Applications* **81**, 103698 (2024).
26. Li, R., Liu, Q. & Liu, L. Novel image encryption algorithm based on improved logistic map. *IET Image Processing* **13**, 125–134 (2019).
27. Iqbal, N. & Hanif, M. An efficient grayscale image encryption scheme based on variable length row-column swapping operations. *Multimedia Tools and Applications* **80**, 36305–36339 (2021).
28. Mohamed, N.A.E.-S., El-Sayed, H. & Youssif, A. Mixed multi-chaos quantum image encryption scheme based on quantum cellular automata (qca). *Fractal and Fractional* **7**, 734 (2023).
29. Qu, G. et al. Optical color image encryption based on hadamard single-pixel imaging and arnold transformation. *Optics and Lasers in Engineering* **137**, 106392 (2021).
30. Hu, W.-W., Zhou, R.-G., Jiang, S., Liu, X. & Luo, J. Quantum image encryption algorithm based on generalized arnold transform and logistic map. *CCF Transactions on High Performance Computing* **2**, 228–253 (2020).
31. Gao, J., Wang, Y., Song, Z. & Wang, S. Quantum image encryption based on quantum dna codec and pixel-level scrambling. *Entropy* **25**, 865 (2023).
32. Zhang, J. et al. Quantum image encryption based on quantum image decomposition. *International journal of theoretical physics* **60**, 2930–2942 (2021).
33. Wen, H., Lin, Y., Kang, S., Zhang, X. & Zou, K. Secure image encryption algorithm using chaos-based block permutation and weighted bit planes chain diffusion. *IScience* **27** (2024).
34. Rehman, A. U. et al. A color image encryption algorithm based on one time key, chaos theory, and concept of rotor machine. *IEEE Access* **8**, 172275–172295 (2020).
35. Kumar, R., Sharma, A. K. et al. Bit-plane based reversible data hiding in encrypted images using multi-level blocking with quad-tree. *IEEE Transactions on Multimedia* (2023).
36. Zhu, S. & Zhu, C. Security analysis and improvement of an image encryption cryptosystem based on bit plane extraction and multi chaos. *Entropy* **23**, 505 (2021).
37. Singh, K. N., Singh, O. P., Baranwal, N. & Singh, A. K. An efficient chaos-based image encryption algorithm using real-time object detection for smart city applications. *Sustainable Energy Technologies and Assessments* **53**, 102566 (2022).
38. Wu, X., Wang, D., Kurths, J. & Kan, H. A novel lossless color image encryption scheme using 2d dwt and 6d hyperchaotic system. *Information Sciences* **349**, 137–153 (2016).
39. Valli, D. & Ganesan, K. Chaos based video encryption using maps and ikeda time delay system. *The European Physical Journal Plus* **132**, 1–18 (2017).
40. Bennett, C. H., Brassard, G. & Ekert, A. K. Quantum cryptography. *Scientific American* **267**, 50–57 (1992).
41. Özkaynak, F. Construction of robust substitution boxes based on chaotic systems. *Neural Computing and Applications* **31**, 3317–3326 (2019).
42. Shafique, A. A new algorithm for the construction of substitution box by using chaotic map. *The European Physical Journal Plus* **135**, 194 (2020).
43. Hussain, I., Shah, T., Gondal, M. A., Khan, W. A. & Mahmood, H. A group theoretic approach to construct cryptographically strong substitution boxes. *Neural Computing and Applications* **23**, 97–104 (2013).
44. Zhou, N.-R., Hu, L.-L., Huang, Z.-W., Wang, M.-M. & Luo, G.-S. Novel multiple color images encryption and decryption scheme based on a bit-level extension algorithm. *Expert Systems with Applications* **238**, 122052 (2024).
45. Alexan, W., Elkandoz, M., Mashaly, M., Azab, E. & Aboshousha, A. Color image encryption through chaos and kaa map. *Ieee Access* **11**, 11541–11554 (2023).
46. Yasser, I., Khalil, A. T., Mohamed, M. A., Samra, A. S. & Khalifa, F. A robust chaos-based technique for medical image encryption. *IEEE Access* **10**, 244–257 (2021).
47. Alvarez, G. & Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. *International journal of bifurcation and chaos* **16**, 2129–2151 (2006).
48. Shafique, A., Mehmood, A., Alawida, M., Khan, A. N. & Khan, A. U. R. A novel machine learning technique for selecting suitable image encryption algorithms for iot applications. *Wireless communications and mobile computing* **2022**, 5108331 (2022).
49. Shannon, C. E. A mathematical theory of communication. *The Bell system technical journal* **27**, 379–423 (1948).

Acknowledgements

This work has been supported by the CHEDDAR: Communications Hub for Empowering Distributed Cloud Computing Applications and Research funded by the UK EPSRC under grant numbers EP/Y037421/1 and EP/X040518/1.

Author contributions

A.S: Conceptualisation, Methodology, Writing - Original Draft; S.A.A.N: Data analysis; A.R: Data curation; M.G: Review and editing; P.P: Theoretical analysis; Q.H.A: Supervision & funding acquisition; M.A.I: Supervision & funding acquisition; J.M: Project administration. All authors reviewed the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to A.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© Crown 2024