scientific reports

Check for updates

OPEN

A lightweight encryption algorithm for resource-constrained IoT devices using quantum and chaotic techniques with metaheuristic optimization

Amer Aljaedi[1]✉, Adel R. Alharbi[1], Abdullah Aljuhni[1], Moahd K. Alghuson[2], Shafi Alassmi[3] & Arslan Shafique[4]✉

As the internet of things (IoT) continues to proliferate, the need for efficient and secure data encryption has become increasingly critical, particularly for resource-constrained devices. Existing encryption methods offer adequate security for digital data; however, they often fall short when applied to resource-constrained IoT devices. This research introduces a novel lightweight encryption algorithm optimized with metaheuristic techniques, incorporating quantum encryption, confusion and diffusion operations, discrete wavelet transform (DWT), and multiple chaotic maps. Initially, a color image is decomposed into its three color components-red (R), green (G), and blue (B)-and then transformed into its quantum representation, where quantum encryption operations are performed. Following this, the quantum image is transformed back into a classical format to apply confusion and diffusion techniques. Confusion is achieved by generating a substitution matrix and applying a modular operation to introduce pixel-level confusion. A key matrix is then created to implement the diffusion operation. In the final phase, DWT is used to extract frequency sub-bands, forming a low-frequency sub-band and further extracting sub-bands up to the 4th level, which are substituted using values from the substitution box. The performance of the proposed encryption framework is evaluated through various statistical analyses, including entropy, correlation, key sensitivity, lossless analysis, and histogram analysis. The results demonstrate notable statistical measures with an entropy of 7.9998, a correlation of 0.0001, and a key space of $2^{947.862}$. Additionally, the encryption's robustness is tested against several cyberattacks, such as noise, cropping, and brute force, showcasing its effectiveness in resisting these threats.

The Internet of Things (IoT) has become a prominent field of research due to its applications across various areas, including smart transportation, healthcare, the environment, infrastructure, and agriculture[1–3]. While definitions of IoT vary depending on the context, it generally refers to a network of interconnected devices with unique IDs that can collect and share data over the Internet, either with or without human involvement. IoT devices, which are central to any IoT application, can be classified into two main categories: resource-rich devices, including smart phones and tablets, and resource-constrained devices, including RFID, sensors, and unmanned aerial vehicles (UAV) storage systems[4]. This research focuses on the second category, which is exponentially gaining popularity due to its widespread application and is expected to significantly increase data exchange as IoT continues to expand.

The proliferation of millions of smart devices across various platforms introduces significant challenges, particularly when transitioning from servers to sensors[5–7]. These challenges include issues related to security and privacy, interoperability, longevity, support, and technology. IoT devices, which interact with the physical world to collect sensitive data or control environmental variables, are particularly vulnerable to security attacks. This makes cybersecurity a major concern for IoT devices, necessitating strict standards for confidentiality,

[1]College of Computing and Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia. [2]Department of Industrial Engineering, Faculty of Engineering, University of Tabuk, Tabuk 71491, Saudi Arabia. [3]Governance Risk and Compliance, NEOM, Tabuk 49643, Saudi Arabia. [4]School of Electronic and Nanoscale Engineering, University of Glasgow, Glasgow G12 8QQ, UK. ✉email: aaljaedi@ut.edu.sa; Arslan.shafique@glasgow.ac.uk

authorization, data integrity, authentication, availability, and regular updates. Figure 1 illustrates these security challenges and requirements.

In this context, cryptography can effectively ensure the authentication, integrity, and confidentiality of data transmitted through IoT devices[8,9]. While conventional PC-based cryptographic algorithms are unsuitable for resource-constrained IoT devices due to their high resource requirements, lightweight cryptography offers a solution by providing a more resource-efficient approach to securing communication on these devices. Many IoT devices, such as RFIDs and sensors, are compact and have limited resources, including small memory, low computing power, and restricted battery life (or no battery in passive RFIDs). These devices often handle real-time applications, requiring fast and accurate responses with the available resources, which poses significant challenges. Designers of IoT devices face risks related to energy capacity and data security. In these situations, conventional cryptography standards may not perform well on IoT devices like RFIDs and sensors. Traditional encryption algorithms such as the Advanced Encryption Standard (AES)[10] and Data Encryption Standard (DES)[11] are widely used for securing data, but they may not be suitable for all applications, particularly those requiring real-time processing with minimal delay. While AES is renowned for its robust security features, due to the multiple encryption rounds, its complexity can be a limitation in scenarios demanding low processing times, such as in unmanned aerial vehicles (UAVs). Moreover, AES consists of several steps, including substitution, permutation, mixing, and key addition. Therefore, the increased number of encryption and mathematically cryptographic encryption operations makes it unsuitable for IoT devices.

However, lightweight cryptography addresses these issues effectively by incorporating features suited for resource-constrained devices, such as minimal memory usage, low processing power, low power consumption, and real-time response capabilities[12–15]. Lightweight cryptography is not only suitable for resource-constrained devices like RFID tags and sensors but is also applicable to more resource-rich devices, such as servers, PCs, tablets, and smartphones, with which these devices interact directly or indirectly.

This research introduces a lightweight cryptographic encryption framework consisting of five key components: (i) converting the plaintext image into its quantum version and then performing quantum encryption; (ii) generating keys using multiple chaotic maps; (iii) applying confusion operations; (iv) applying diffusion operations; and (v) using the discrete wavelet transform (DWT). According to existing studies, creating diffusion in digital images of size 256×256 with substitution boxes (S-boxes) takes approximately 11 seconds on a system with 8GB RAM, a 512GB SSD, Windows 11, and a Core i5 processor. Additionally, the time required for the S-box encryption algorithm is influenced by the size of the input image. This can be summarized in two main points:

- For larger images, such as 512×512, the encryption process with S-boxes will take more time.
- With the same input image size, insufficient hardware specifications will also increase the encryption time. To reduce computational time, either the input image size must be smaller or the hardware must be more efficient. Hence, there is a tradeoff between image size and hardware specifications.

The proposed encryption algorithm overcomes this trade-off. It maintains efficiency even with larger images and the specified hardware, thanks to the incorporation of DWT. This capability will be detailed in the subsequent sections, where the encryption process is thoroughly explained.

Motivation and novelty

As the Internet of Things (IoT) ecosystem continues to expand, securing sensitive data in resource-constrained devices has become a critical challenge[16]. Traditional encryption methods, while effective for general computing, often introduce excessive computational overhead. This makes them non-practical for low-power IoT environments. The increasing number of cyberattacks, including brute-force, differential, and noise-

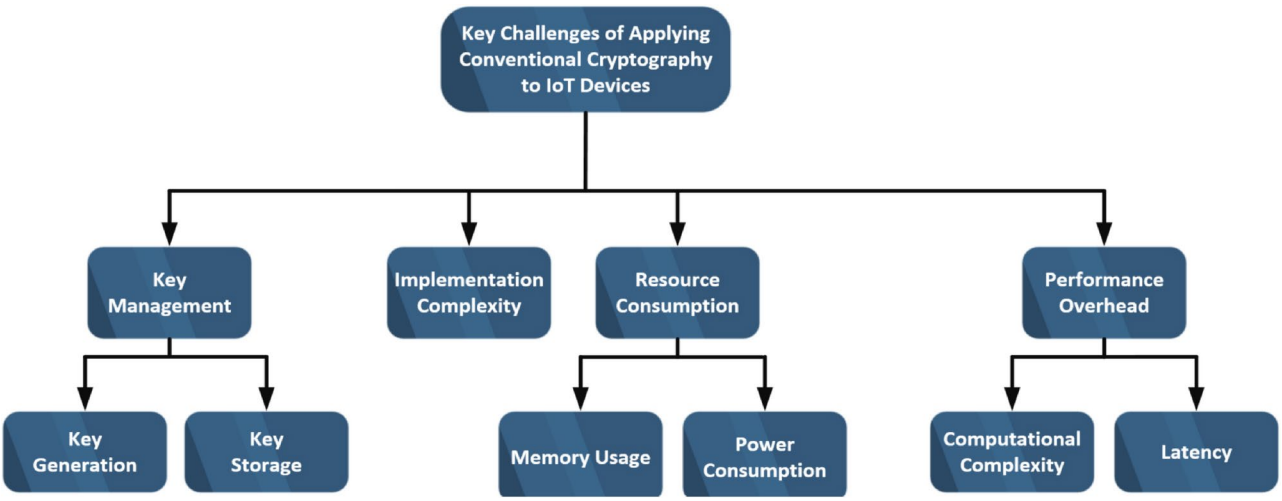


Fig. 1. Challenges of conventional cryptography in IoT devices.

based attacks, further necessitates the development of lightweight yet highly secure encryption frameworks. Additionally, the emergence of quantum computing threats shows the need for encryption schemes that offer long-term security resilience. Motivated by these critical challenges, this research proposed a lightweight quantum-chaotic encryption for IoT designed to achieve an optimal balance between security, efficiency, and practical applicability in IoT environments.

The novelty of this research work lies in its unique integration of quantum encryption, chaotic systems, metaheuristic optimization, and DWT techniques. Unlike conventional methods that rely only on chaotic maps or classical cryptography, this approach transforms data into the quantum domain and applies the quantum operations to overcome the challenges that can raised by the quantum threats. The inclusion of metaheuristic optimization also enhances key generation whihc increases randomness and making the encryption scheme more resistant to brute-force attacks. Furthermore, DWT-based substitution enables multi-resolution encryption which ensures the additional layers of security against statistical and cryptographic attacks. These advancements introduce a new standard in lightweight encryption, enhancing the proposed encryption framework's adaptability to future cryptographic challenges, including quantum attacks.

Organization of the paper

The remainder of this paper is organized as follows: Section Related work provides an overview of existing encryption frameworks, highlighting their advantages, shortcomings, and potential solutions. Section Preliminaries covers the preliminary knowledge necessary for developing the proposed encryption framework. Section Proposed encryption framework provides a detailed explanation of the steps involved and the mathematics required to develop the proposed framework. Section Experimental result and analysis presents experimental results and analyzes the effectiveness of the proposed method compared to existing encryption schemes. Finally, Section Conclusion concludes the proposed research.

Related work

Internet of Things (IoT) faces challenges due to limited energy, memory, and computing power[17–19]. Due to such limitations, IoT devices are highly susceptible to cyberattacks. In the present high-tech era, vast amounts of data are transmitted over the insecure channel of the Internet[20]. Many existing encryption schemes are vulnerable to cyberattacks such as brute force and known/chosen plaintext attacks[21–26]. Therefore, there is a need to develop a lightweight encryption scheme that not only protects digital data transmitted over insecure channels but also ensures time efficiency for real-time applications. In[27], Hedayati et al. proposed a lightweight image encryption scheme specially designed for IoT devices. The authors incorporated scan-based block compression and selective pixel encryption to make it time-efficient. Their method encrypts image data in a single round which also helps to minimize the encryption computational time. Tests on an IoT testbed demonstrated that their approach reduces power consumption by 15% and packet rate by 26% compared to existing methods. However, it is susceptible to attacks targeting the reduced complexity of its encryption process. Additionally, the selective pixel encryption approach leaves some parts of the image which can be the source of information leakage as well. In[28], Ince et al. proposed a new Corner Traversal algorithm to improve pixel scrambling by providing enhanced confusion and efficiency with lower time complexity. When integrated with chaos-based diffusion, it creates a robust encryption scheme. The results suggest it meets the security needs of IoT systems. However, vulnerabilities in its confusion phase and the added complexity of chaos-based methods make the encryption algorithm susceptible to quantum and highly advanced cryptographic attacks such as Quantum memory attacks, Quantum Key Distribution (QKD) eavesdropping.

In[29], Biswas et al. developed LRBC, a lightweight encryption method for IoT devices that enhances data security using substitution-permutation networks (SPN)[30] and Feistel structures[31]. Efficient and tested on FPGA and ASIC chips, it consumes only 11.40 μw of power and occupies 258.9 GE. Security analyses show high robustness, with an average avalanche effect of 58% for plaintext and 55.75% for keys. However, LRBC may be susceptible to attacks targeting its SPN and Feistel components, and its resilience against advanced threats has not been fully explored. In[9], Gupta et al. developed a two-layer encryption scheme that uses discrete wavelet transform (DWT) watermarking combined with a hybrid encryption technique featuring a logistic chaotic map[32] and crossover operation[33]. This approach, which encrypts a secret image as a watermark and generates random session keys, offers enhanced security and performance, achieving high NPCR (99.63) and information entropy (7.9973). However, the scheme may be vulnerable to attacks exploiting weaknesses in the logistic chaotic map and crossover methods, and DWT-based watermarking might be susceptible to advanced image manipulation. In[34], Diro et al. developed a resource-efficient end-to-end security scheme for IoT that offloads security tasks to nearby fog nodes and uses symmetric-key payload encryption. This approach reduces communication overhead, requires fewer handshakes, and transmits smaller messages (184 bytes vs. 332 bytes for TLS), outperforming TLS in resource usage while maintaining equivalent authentication. However, it may be vulnerable if fog nodes are compromised or poorly secured, and the symmetric-key encryption could be less secure if key management is not robust. In[35], kanwal et al., proposed an IoT-Blockchain system with chaos encryption to enhance the security of medical data transactions. While it effectively prevents data breaches and ensures integrity, challenges include computational overhead, scalability issues, and susceptibility to advanced cryptanalysis attacks. In[36], Inam et al., introduced a Cycle_GAN-based encryption method for medical images in IoMT, ensuring secure data transformation without requiring paired training data. It enhances confidentiality and robustness but faces challenges such as vulnerability to adversarial attacks and high computational demands affecting real-time processing. In[37], Inam et al., presented BCAES, a Blockchain-based Chaotic Arnold's Cat Map Encryption Scheme, to secure medical image processing. It ensures data integrity and protection but faces challenges in computational efficiency and scalability. Optimizing encryption speed and reducing blockchain storage overhead could improve its real-time applicability. In[38], Jain et al. developed Multiple Map Chaos Based

Image Encryption (MMCBIE), a novel method for IoT that employs multiple chaotic maps, including Henon and 2D-Logistic Chaotic Transforms, to enhance encryption robustness. MMCBIE achieves high security and performance, with strong evaluation scores such as an NPCR of 99.603 and UACI of 32.8828, making it nearly indistinguishable from visual noise. However, its effectiveness could be compromised if the chaotic maps are not complex enough or are vulnerable to advanced attacks, and any implementation flaws could impact its security. In[39], Trujillo et al. proposed an encryption scheme based on embedded system that uses four chaotic maps to create a pseudo-random number generator (PRNG). This PRNG is employed in a simple encryption algorithm for real-time RGB images in a machine-to-machine (M2M) environment. However, the system is suspectable if the chaotic maps or PRNG lack sufficient randomness or predictability, and the use of MQTT over public networks could expose it to interception and tampering.

In[40], Gabr et al., proposed a new image encryption algorithm that leverages chaotic and hyper-chaotic systems[41], specifically the Chua and Chen models, to enhance security through rescaling, rotation, and randomization. With a large key space, the algorithm resists brute-force attacks and is efficient for real-time use. While testing confirms its robustness, challenges include sensitivity to initial conditions and susceptibility to advanced cryptanalysis if weaknesses exist in its transformations. In[42], Alexan et al., presented a 5-stage image encryption algorithm that enhances security using chaos-based transformations, XOR operations, S-box substitution, and pixel scrambling. It demonstrates resilience against various attacks and passes the NIST SP 800-22 test, making it suitable for real-time secure transmission. However, reliance on predefined chaotic maps may introduce predictability, and further optimization is needed to strengthen resistance against advanced cryptanalysis. In[43], Alexan et al., presented a 3-stage image encryption algorithm using a fractional-order 4D Chen system, DFT-based DNA coding, an S-box, and a Mersenne Twister key. It offers strong security with high entropy, zero correlation, and passes NIST SP 800 tests, featuring a large key space (2^{1754}) and a fast encryption rate (72.6 Mbps). However, its computational complexity may limit real-time use on resource-constrained devices, and further analysis is needed to ensure resistance against advanced cryptanalysis. In[44], El et al., introduced an image encryption algorithm for securing color medical images in cloud storage, utilizing Fibonacci Q-matrices, an S-box transformation, and a hyperchaotic key for enhanced security. The algorithm offers a high encryption rate (16.65 Mbps), making it suitable for medical imaging applications like CT scans. In[45], et al., Youssef et al. presented an image encryption scheme integrating hyperchaotic systems, SVD, RC5 operations, permutation, and an XORshift-based S-box to secure satellite imagery. By merging multiple images into a single augmented image, the method prevents traffic analysis attacks. In[46], Clemente et al. developed a chaos-based lightweight encryption scheme for IoT healthcare systems, particularly for wearable devices. Their proposed scheme used a 2D 4-scroll chaotic attractor to enhance data security. The scheme, tested on an ARM-based microcontroller to show the effectiveness. However, it is vulnerable to the chaotic system's parameters that are not robust. Moreover, it faces hardware-specific attacks or limitations due to its ARM-based implementation. In[47], Mondal et al. proposed a lightweight image encryption technique using chaotic maps and diffusion circuits[48] to generate random sequences for pixel permutation and substitution These all operations are executed in a single scan to reduce time complexity. While it minimizes computational overhead with simple bit-wise operations, the scheme is also vulnerable if the chaotic maps lack complexity or predictability. Table 1 provides a summary of existing lightweight encryption schemes for IoT devices.

Among the key challenges identified in this literature review are strong data security and high computational complexity. In an IoT environment, minimizing computational time and memory usage is crucial. To address the vulnerabilities outlined in Table 1, the following key contributions have been made in this research.

Methodology name	Application domain	Real-world performance	Robustness against attacks	Disadvantages	Potential solutions
Hedayati et al.[27]	IoT encryption	15% power saved	Weak to simple attacks	Information leakage risk	Increase encryption strength
Ince et al.[28]	Pixel scrambling	Low time complexity	Quantum attack risk	Weak confusion phase	Strengthen confusion layer
Biswas et al.[29]	SPN-Feistel encryption	11.4 mW power	SPN/Feistel attack risk	Advanced threats unknown	Enhance resilience measures
Gupta et al.[9]	Hybrid encryption	High entropy score	Chaotic method weakness	DWT watermarking flaw	Improve chaotic models
Diro et al.[34]	Secure IoT	Reduced overhead	Vulnerable fog nodes	Weak key management	Strengthen key handling
Gabr et al.[40]	Digital security	Large key space	Protects from brute force	Initial sensitivity risk	Better key management
Alexan et al.[42]	Secure networks	NIST SP 800-22 pass	Resists brute-force	Chaotic map predictability	Improve randomness
Alexan et al.[43]	Digital encryption	72.6 Mbps speed	High entropy	High complexity	Optimize computation
El et al.[44]	Medical imaging	16.65 Mbps rate	Ensures confidentiality	High computation cost	Reduce complexity
Youssef et al.[45]	Satellite security	Prevents traffic attacks	Large key space	Computational overhead	Optimize encryption stages
Jain et al.[38]	Chaotic encryption	High NPCR score	Weak chaotic maps	Poor confusion phase	Use stronger chaos
Trujillo et al.[39]	Embedded systems	Real-time RGB use	Low randomness	MQTT risk	Secure MQTT layer
Clemente et al.[46]	Healthcare security	ARM microcontroller	Weak chaos parameters	Hardware limits	Improve chaos tuning
Mondal et al.[47]	Chaotic maps	Low overhead	Simple chaos maps	Weak confusion	Increase complexity
Kanwal et al.[35]	IoT blockchain	Prevents breaches	Cryptanalysis risk	High cost	Enhance efficiency
Inam et al.[36]	Medical images	Secure transformation	Adversarial attack risk	High cost	Improve defenses
Inam et al.[37]	Blockchain security	Ensures integrity	Performance issues	Storage overhead	Optimize processing

Table 1. Summary of existing lightweight encryption schemes for IoT devices.

- In this research, a new encryption algorithm is proposed and optimised for resource-constrained IoT devices to address the limitations of traditional methods in terms of computational and energy efficiency.
- The proposed algorithm combines multiple cryptographic techniques such as quantum encryption, chaotic systems, and DWT to develop a hybrid approach that enhances both security and efficiency.
- The proposed encryption framework is fine-tuned using metaheuristic optimisation techniques to balance performance and security. Moreover, to provide an extra layer of security to the digital images, the proposed framework incorporates quantum encryption, which enhances the robustness against quantum attacks.
- The discrete wavelet transform (DWT) is used to extract frequency sub-bands from the image, with only the low-frequency band considered for encryption. This reduces computational complexity while maintaining strong security, making it well-suited for real-time IoT applications.

Preliminaries

This section provides a brief explanation of the preliminary knowledge that is required to develop the proposed encryption framework.

7D hyperchaotic system

In[49], Yang et al. introduced a seven-dimensional hyperchaotic system (7DHCS), which is derived from combining a 6D hyperchaotic system[50] with a 1D linear system[51]. This system features a complex structure that enhances its hyperchaotic behavior compared to existing chaotic systems[52]. Its intricate dynamic properties make it suitable for secure communications. The system can be mathematically defined using Eq. 1.

$$\begin{cases} d_1^{'} = L(d_2 - d_1) + d_4 + nd_6 \\ d_2^{'} = md_1 - d_2d_1d_3 + d_5 \\ d_3^{'} = -od_3 + d_1d_2 \\ d_4^{'} = pd_4 - d_1d_3 \\ d_5^{'} = -xd_2 + d_6 \\ d_6^{'} = u_1d_1 + u_2u_2 \\ d_7^{'} = kd_7 + vd_4 \end{cases} \tag{1}$$

Here, the initial states are represented by $d_1, d_2, d_3, d_4, d_5, d_6,$ and d_7. Control parameters are denoted by $n, p, x, u_1, u_2,$ and k. The constant parameters of System (1) are $L, o,$ and m, with v representing the coupling parameter. The values $d_1, d_2, d_3, d_4, d_5, d_6,$ and d_7 are derived from random sequences. Figure 2a and b illustrate the hyperchaotic attractor of the 7DHS system with $L = 9, n = 0.9, m = 26, o = 2.66, p = 2, x = 9.7, u_1 = 0.8, u_2 = 1.98, k = 0.99,$ and $v = 0.97$ in the $d_2 - d_5 - d_6$ and $d_1 - d_2 - d_6$ spaces, respectively.

Discrete wavelet transform

The Wavelet Transform (WT) is a mathematical technique used to efficiently analyze signals with varying frequencies[53,54]. It breaks down a signal into different components by applying shifts and scales to a mother wavelet, which allows the signal to be segmented into various wavelets. These wavelets can then be processed further, including through decimation to discard finer details, thus isolating specific sub-bands: HL, HH, LL, and LH. In image processing, the LL sub-band typically contains the majority of the image's main information, while the high-frequency sub-bands capture finer details like edges as shown in Fig. 3.

The proposed encryption method utilizes the Haar transform, represented by $G' = WGW^T$, where G is an $A \times A$ image matrix, W is the Haar transform matrix, and G' is the resulting transformed image matrix. This

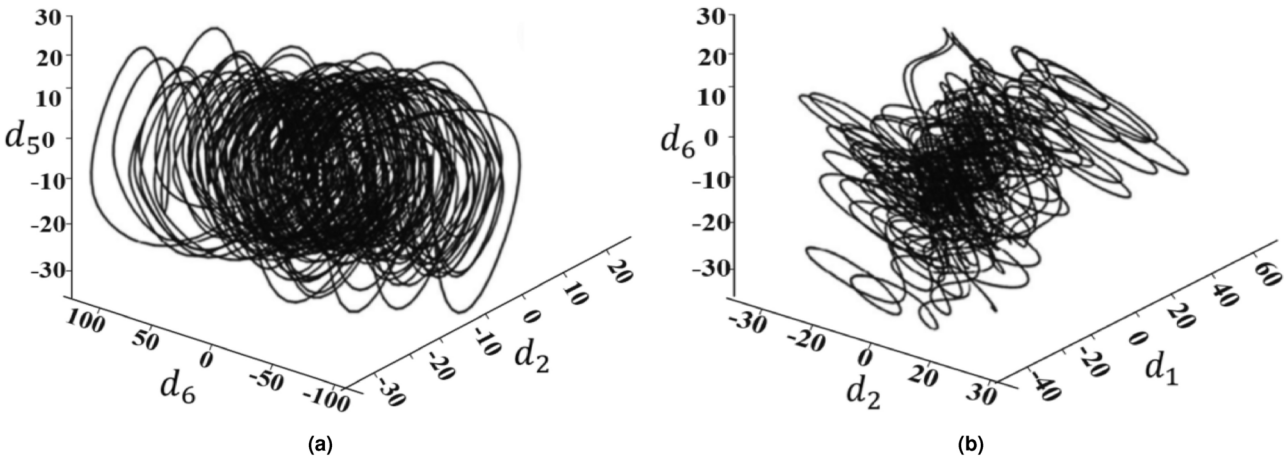
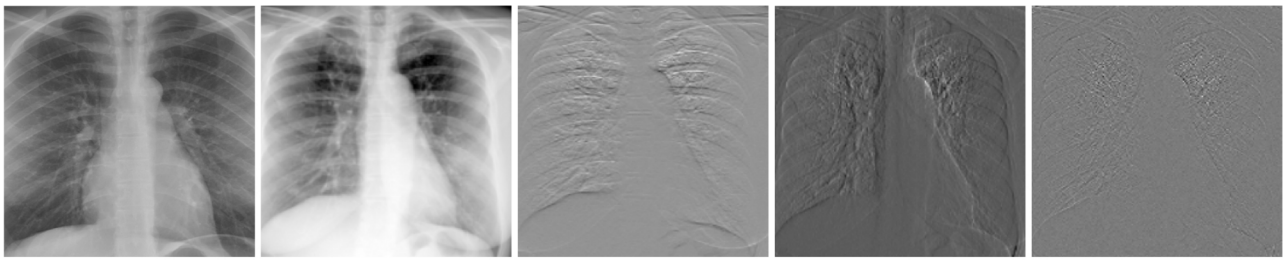


Fig. 2. Hyperchaotic attractor: (**a**) $d_2 - d_5 - d_6$ (**b**) $d_1 - d_2 - d_6$ space.



(a) Plaintext Xray image: 256 × 256 **(b)** LL sub-band: 128 × 128 **(c)** LH sub-band: 128 × 128 **(d)** HL sub-band: 128 × 128 **(e)** HH sub-band: 128 × 128

Fig. 3. Extraction of frequency bands from Xray image using DWT.

process relies on the Haar basis function $g_m(z)$, which is defined over the interval $z \in [0, 1]$ with m ranging from 0 to $M - 1$. The decomposition of this basis function is a key aspect of the transformation.

The Haar wavelet transformation is expressed through a matrix equation where G' is the transformed image of size $A \times A$, W is the Haar transformation matrix of the same dimensions, and G is the resulting transformed matrix, also $A \times A$. This matrix includes the Haar basis function $g_m(z)$, defined over the interval $z \in [0, 1]$ with m ranging from 0 to $M - 1$. The following explanation details the breakdown of this function.

$$p = 2^y + u \tag{2}$$

Here, y represents the highest power of 2 within the integer p, and u denotes the remainder, defined as $u = 2^y - p$. Equation 3 formally defines the Haar basis function.

$$h_f(c) = \frac{1}{\sqrt{S}} \begin{cases} 1 & if p = 0 \quad \& \quad 0 \le c \le 1 \\ 2^{y/2} & if x > 0 \quad \& \quad u/2^y \le c < \frac{u+0.5}{2^u} \\ -2^{y/2} & if p > 0 \quad \& \quad (u+0.5)/2^y \le c < \frac{u+1}{2^y} \\ 0 & \text{Elsewhere} \end{cases} \tag{3}$$

The matrix required for performing the 2D discrete Haar wavelet transform can be derived by using the reciprocal transformation kernel provided in Eq. 4.

$$h'(c, p) = \frac{1}{\sqrt{N}} h_p(c/N) \qquad for \;\; c = 0, 1, 2, \cdots, K - 1 \tag{4}$$

where $h(c, p)$ will be:

$$h(c, p) = H' = \begin{bmatrix} h_0(\frac{0}{N}) & h_0(\frac{1}{N}) & \cdots & h_0(\frac{N-1}{N}) \\ h_1(\frac{0}{N}) & h_1(\frac{1}{N}) & \cdots & h_1(\frac{N-1}{N}) \\ h_2(\frac{0}{N}) & h_2(\frac{1}{N}) & \cdots & h_2(\frac{N-1}{N}) \\ \vdots & \vdots & \ddots & \vdots \\ h_{N-1}(\frac{0}{N}) & h_{N-1}(\frac{1}{N}) & \cdots & h_{N-1}(\frac{N-1}{N}) \end{bmatrix} \tag{5}$$

Proposed encryption framework

This research proposes a lightweight cryptographic encryption framework tailored for resource-constrained IoT devices, comprising five key components:

1. Generating keys through multiple chaotic maps
2. Converting the plaintext image into a quantum version and performing quantum encryption
3. Implementing confusion operations
4. Applying diffusion operations
5. Employing the discrete wavelet transform (DWT).

The block diagram of the proposed encryption framework is displayed in Fig. 4.

Key generation

In this proposed research, chaotic key sequences are generated by integrating 7DHCS and the chaotic sine map. To enhance the randomness of these sequences, appropriate values for initial conditions and control parameters are chosen from the chaotic ranges of both maps. These sequences are then utilized to introduce confusion into the plaintext image. The algorithm for generating these random sequences is outlined in Algorithm 1.

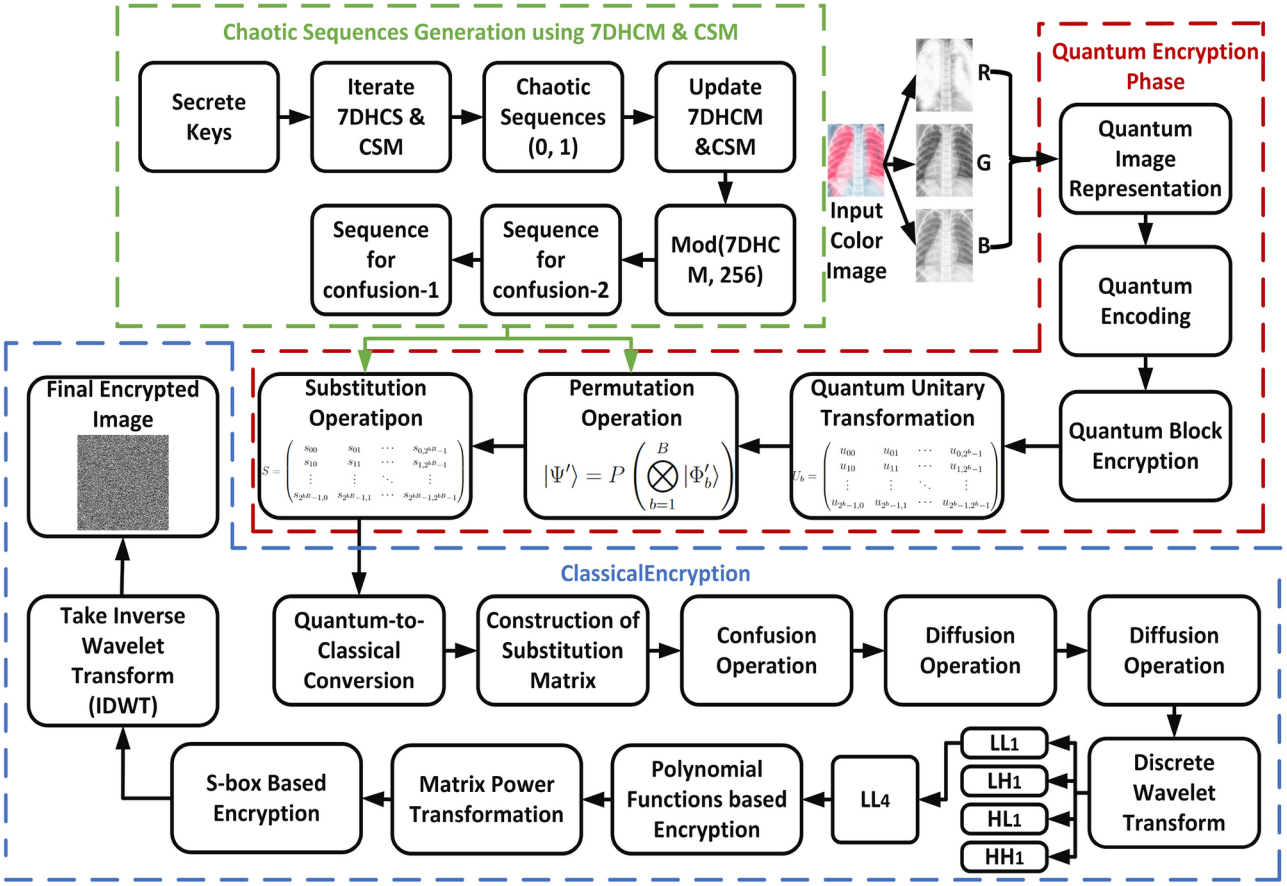


Fig. 4. Proposed encryption framework.

```
function GENERATE_COMBINED_CHAOTIC_SEQUENCE(initial_conditions, parameters, phi, num_steps)
    d ← initial_conditions  // Initial state for 7D map
    L, n, m, o, p, x, u1, u2, k, v ← parameters
    xi ← initial_conditions[0]  // Initial state for sine map
    sequence_7D ← empty list
    sequence_sine ← empty list
    for step from 1 to num_steps do
        // Update 7D chaotic map
        d1, d2, d3, d4, d5, d6, d7 ← d
        for i in range(1, 8) do
            if i == 1 then
                d1_new ← L * (d2 − d1) + d4 + n * d6
            else if i == 2 then
                d2_new ← m * d1 − d2 * d1 * d3 + d5
            else if i == 3 then
                d3_new ← −o * d3 + d1 * d2
            else if i == 4 then
                d4_new ← p * d4 − d1 * d3
            else if i == 5 then
                d5_new ← −x * d2 + d6
            else if i == 6 then
                d6_new ← u1 * d1 + u2 * d2
            else if i == 7 then
                d7_new ← k * d7 + v * d4
            end if
        end for
        d ← [d1_new, d2_new, d3_new, d4_new, d5_new, d6_new, d7_new]
        sequence_7D.append(d1_new)
        // Update chaotic sine map (ξ_{n+1} = φ sin(πξ_n))
        for j in range(1, 3) do
            if j == 1 then
                xi_new ← phi * sin(π * xi)
            else if j == 2 then
                do some other operations if needed
            end if
        end for
        xi ← xi_new
        sequence_sine.append(xi_new)
    end for
    return sequence_7D, sequence_sine
end function
```

Algorithm 1. Generate chaotic sequences from 7D chaotic map and chaotic sine map

Quantum image representation

Input color image image (I_g) of size $M \times N \times 3$ represented as a matrix of pixel values $p_{i,j}$ where $i \in \{1, \ldots, M\}$ and $j \in \{1, \ldots, N\}$. For image normalization, Eq. 6 is utilized.

$$\tilde{p}_{i,j} = \frac{p_{i,j}}{255} \tag{6}$$

where $\tilde{p}_{i,j}$ is the normalized pixel value in the range $[0, 1]$.

Quantum encoding

To encode an image into a quantum state, each pixel value $p_{i,j}$ is first normalized to $\tilde{p}_{i,j}$, ensuring that all pixel values lie within the range [0,1]. Each normalized pixel value is then represented using k qubits.

A common approach for quantum encoding is amplitude encoding, which maps a classical pixel value $\tilde{p}_{i,j}$ to the probability amplitude of a quantum state. This encoding is performed as follows:

$$|\psi_{i,j}\rangle = \sqrt{\tilde{p}_{i,j}}|0\rangle + \sqrt{1-\tilde{p}_{i,j}}|1\rangle \tag{7}$$

Here, $|\psi_{i,j}\rangle$ represents the quantum state corresponding to a single pixel. The coefficients $\sqrt{\tilde{p}_{i,j}}$ and $\sqrt{1-\tilde{p}_{i,j}}$ ensure that the quantum state remains normalized, i.e., the sum of squared amplitudes is equal to 1.

Encoding an entire image
For an image I_g of size $M \times N \times 3$, where each pixel is independently encoded into quantum states, the overall quantum representation is constructed by taking the tensor product of all pixel states:

$$|\Psi\rangle = \bigotimes_{i=1}^{M} \bigotimes_{j=1}^{N} |\psi_{i,j}\rangle \tag{8}$$

The $\bigotimes$ symbol denotes the tensor product, which is used to combine multiple quantum states into a larger quantum system. The right-hand side of the equation represents the complete quantum state of the image, where each pixel (i, j) contributes an independent quantum state $|\psi_{i,j}\rangle$. The tensor product operation ensures that all pixel states are combined into a single quantum representation.

Since each pixel value is encoded using k qubits, the total quantum state of the image spans an $M \times N \times k$ -dimensional Hilbert space, represented as:

$$|\Psi\rangle = \bigotimes_{i=1}^{M} \bigotimes_{j=1}^{N} \left(\sqrt{\tilde{p}_{i,j}}|0\rangle^{(k)} + \sqrt{1-\tilde{p}_{i,j}}|1\rangle^{(k)}\right) \tag{9}$$

Here, $|0\rangle^{(k)}$ and $|1\rangle^{(k)}$ represent the k-qubit basis states used for encoding each pixel value. This formulation allows the entire image to be represented as a high-dimensional quantum state, enabling further quantum processing such as encryption.

Quantum block encryption

Divide the quantum state $|\Psi\rangle$ as mentioned in Eq. 9 into B blocks, each containing b qubits. Let $|\Phi_b\rangle$ represent the state of each block. For block b, $|\Phi_b\rangle = \bigotimes_{l=1}^{b} |\phi_l\rangle$, where $|\phi_l\rangle$ denotes each qubit within the block.

Unitary transformation
Apply a unitary transformation U_b to each block $|\Phi_b\rangle$ using Eq. 10.

$$|\Phi_b'\rangle = U_b|\Phi_b\rangle \tag{10}$$

where U_b is a unitary matrix of size $2^b \times 2^b$, as mentioned in Eq. 11.

$$U_b = \begin{pmatrix} u_{00} & u_{01} & \cdots & u_{0,2^b-1} \\ u_{10} & u_{11} & \cdots & u_{1,2^b-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{2^b-1,0} & u_{2^b-1,1} & \cdots & u_{2^b-1,2^b-1} \end{pmatrix} \tag{11}$$

Applying U_b results in:$|\Phi_b'\rangle = \sum_{j=0}^{2^b-1} \sum_{i=0}^{2^b-1} u_{ij}|i\rangle\langle j||\Phi_b\rangle$.

Permutation operation
Apply a permutation operation P to reorder the blocks. Equation 12 illustrates the state after applying the permutation matrix P, which rearranges the B blocks.

$$|\Psi'\rangle = P\left(\bigotimes_{b=1}^{B} |\Phi_b'\rangle\right) \tag{12}$$

where P is a permutation matrix of dimension $2^{bB} \times 2^{bB}$, P acts on the entire block system, and $\bigotimes_{b=1}^{B} |\Phi_b'\rangle$ represents the tensor product of all blocks before permutation.

Quantum substitution-permutation network (QSPN)

The $|\Psi'\rangle$ is obtained in the previous step (Eq. 12) and will undergo substitution and then permutation operations.

Substitution (S)
In the proposed encryption framework, a substitution operation is applied to a quantum state to substitute each basis state with another basis state according to a predefined rule. The predefined rule is the substitution operation is defined by a unitary matrix S. Each element S_{ij} of this matrix specifies the amplitude transformation from basis state $|j\rangle$ to basis state $|i\rangle$.

Let S be a unitary matrix that operates on the entire quantum state or on individual blocks. Tbhe substitution operation is defined in Eq. 13.

$$|\Psi''\rangle = S|\Psi'\rangle \tag{13}$$

where S is a unitary matrix of dimension $2^{bB} \times 2^{bB}$. The unitary matrix S might be:

$$S = \begin{pmatrix} s_{00} & s_{01} & \cdots & s_{0,2^{bB}-1} \\ s_{10} & s_{11} & \cdots & s_{1,2^{bB}-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{2^{bB}-1,0} & s_{2^{bB}-1,1} & \cdots & s_{2^{bB}-1,2^{bB}-1} \end{pmatrix}$$

Permutation (P)
Apply a permutation operation P, which rearranges qubits or blocks. For a permutation matrix P:

$$|\Psi'''\rangle = P|\Psi''\rangle$$

where P is a permutation matrix with dimensions appropriate to the state size. Repeat the substitution and permutation steps for R rounds. The final state is obtained by using Eq. 14:

$$|\Psi_f\rangle = P_R\left(S_R\left(\cdots P_1\left(S_1|\Psi\rangle\right)\cdots\right)\right) \tag{14}$$

where S_i and P_i denote the substitution and permutation operations in the i-th round.

Quantum-to-classical conversion
Measure the quantum state $|\Psi_f\rangle$ in the computational basis. For each qubit $q_{i,j}$ in the state, the measurement outcomes $m_{i,j}$ are binary strings. Let's denote the measurement outcome for a qubit as $m_{i,j}$ where $m_{i,j}$ is a k -bit string as given in Eq. 15.

$$m_{i,j} = \text{Measure}(|\psi'_{i,j}\rangle) \tag{15}$$

Convert binary measurement outcomes to grayscale pixel values. If $m_{i,j}$ is a binary string, convert it to a decimal number and then to a pixel value according to Eq. 16.

$$T_{i,j} = \text{GrayScaleFromBinary}(m_{i,j}) \tag{16}$$

where $\text{GrayScaleFromBinary}$ maps the binary string $m_{i,j}$ to the original grayscale range $[0, 255]$.

Confusion and diffusion operations

Consider a matrix $T_{i,j}$ of size $m \times n$ where $T_{i,j}$ represents the plaintext matrix. The confusion operation can be implemented using the random sequences generated by the 7D chaotic map and the chaotic sine map.

The 7D chaotic map generates a sequence seq_{7D} which we denote as:$\text{seq}_{7D} = [d_1, d_2, \ldots, d_{mn}]$. Where each d_i is derived from iterating through the 7D chaotic map equations (Eq. 1). Moreover, the chaotic sine map generates a sequence seq_{sine} is $\text{seq}_{sine} = [s_1, s_2, \ldots, s_{mn}]$. Where s_n is calculated iteratively using the sine map equation (Eq. 2). To combine both the sequences, Eq. 17 is utilized.

$$\text{seq}_{combined}[i] = (\text{seq}_{7D}[i] \oplus \text{seq}_{sine}[i]) \mod 2^b \tag{17}$$

where $\oplus$ denotes element-wise XOR and b is the bit-length of each element.

Construct substitution matrix S
The substitution matrix S is derived from the combined sequence as given in Eq. 18.

$$S_{i,j} = (\text{seq}_{combined}[(i-1)\cdot n + (j-1)] + f(d_i, s_j)) \mod 2^b \tag{18}$$

where $f(d_i, s_j)$ is a nonlinear function:$f(d_i, s_j) = \left\lfloor \frac{d_i \cdot s_j}{2^b} \right\rfloor$, and d_i and s_j are the chaotic values from the sequences.

Confusion operation
The confusion operation is applied using Eq. 19.

$$T'_{i,j} = (T_{i,j} + S_{i,j} + g(d_i, s_j)) \mod 2^b \tag{19}$$

where $g(d_i, s_j)$ is another nonlinear function which is defined as $g(d_i, s_j) = \left(d_i^2 \oplus s_j^2\right) \mod 2^b$.

Diffusion operation
To apply the diffusion operation, first a key matrix *K* is constructed.

The matrix K is generated by combining the chaotic sequences obtained from both the 7D chaotic map and the chaotic sine map using Eq. 20.

$$\text{seq}_{combined}[i] = (\text{seq}_{7D}[i] \oplus \text{seq}_{sine}[i]) \mod 2^b \tag{20}$$

where $\oplus$ denotes the XOR operation and b is the bit-length of each sequence element.

For the matrix K, a nonlinear function $h(d_i, s_j)$ is used to add additional complexity to the key matrix generation. This function is defined in Eq. 21.

$$h(d_i, s_j) = \left\lfloor \frac{d_i + s_j}{2} \right\rfloor \mod 2^b \tag{21}$$

where d_i and s_j are elements from the 7D chaotic map sequence and the chaotic sine map sequence, respectively. The final key matrix K is constructed using Eq. 22.

$$K_{i,j} = (\text{seq}_{combined}[(i-1) \cdot n + (j-1)] + h(d_i, s_j)) \mod 2^b \tag{22}$$

where $\text{seq}_{combined}[(i-1) \cdot n + (j-1)]$ is the combined sequence value for the position (i, j), and $h(d_i, s_j)$ introduces a nonlinear modification based on the chaotic sequences.

After generating $K_{i,j}$, a diffusion transformation is applied according to Eq. 23.

$$D''_{i,j} = \left(\sum_{k=1}^{n} K_{i,k} \cdot T'_{k,j} + j \cdot \left\lfloor \frac{K_{i,j}}{2} \right\rfloor \right) \mod 2^b \tag{23}$$

where the summation involves matrix multiplication, and an additional term $j \cdot \left\lfloor \frac{K_{i,j}}{2} \right\rfloor$ introduces further complexity.

Incorporation of DWT

Now to encrypt $D''_{i,j}$, a linear and nonlinear transformation is applied after extracting frequency subbands from $D''_{i,j}$ using DWT. The DWT decomposes an image matrix into different frequency subbands. For a given image matrix $T''_{i,j}$ of size $M \times N$, the DWT can be applied to obtain:

- LL_1: Low-Low subband (approximation coefficients)
- LH_1: Low-High subband (horizontal detail coefficients)
- HL_1: High-Low subband (vertical detail coefficients)
- HH_1: High-High subband (diagonal detail coefficients)

For an image matrix D'', the DWT decomposition can be mathematically expressed in Eq. 24.

$$D'' = \text{DWT}(T''_{i,j}) = \begin{bmatrix} LL_1 & LH_1 \\ HL_1 & HH_1 \end{bmatrix} \tag{24}$$

The incorporation of DWT enhances the encryption process by introducing frequency-based security measures. Since the approximation coefficients (LL_4) retain most of the image energy, applying encryption to these subbands ensures that the most critical information is effectively protected. Additionally, this frequency-domain transformation increases the sensitivity of the encryption scheme, making it highly resistant to differential and statistical attacks.

Where each subband is obtained through filtering operations with wavelet filters ψ and ϕ for approximation and detail using Eq. 25.

$$\begin{cases} LL_1 = \text{DownSample}(\text{Convolve}(D''_{i,j}, \phi) \cdot \text{Convolve}(D''_{i,j}, \phi)) \\ LH_1 = \text{DownSample}(\text{Convolve}(D''_{i,j}, \phi) \cdot \text{Convolve}(D''_{i,j}, \psi)) \\ HL_1 = \text{DownSample}(\text{Convolve}(D''_{i,j}, \psi) \cdot \text{Convolve}(D''_{i,j}, \phi)) \\ HH_1 = \text{DownSample}(\text{Convolve}(D''_{i,j}, \psi) \cdot \text{Convolve}(D''_{i,j}, \psi)) \end{cases} \tag{25}$$

Furthermore, the use of DWT improves key sensitivity by ensuring that even minor modifications in encryption parameters or input images lead to substantial variations in the transformed coefficients. This significantly strengthens the diffusion property of the encryption scheme. Additionally, because the encryption is applied at the frequency level, attackers attempting direct pixel-based statistical analysis will find it significantly harder to retrieve useful information.

Before substitution, apply linear and nonlinear transformations on LL_4 for enhanced security. The linear transformation is applied using Eq. 26.

$$LL_4^{'} = A \cdot LL_4 \cdot A^T \tag{26}$$

By encrypting the LL_4 coefficients rather than the raw pixel values, the algorithm achieves a higher level of security while maintaining efficiency. Since the approximation subband represents the most important structural

information of the image, tampering with it ensures that decryption without the correct key becomes practically infeasible.

Polynomial function
Apply a polynomial function to each element of the matrix using Eq. 27.

$$LL''_{4ij} = (LL'_{4ij})^2 + \mathrm{Exp}(LL'_{4ij}) \mod 2^b \tag{27}$$

Matrix power function
A matrix power function is applied using Eq. 28.

$$LL''_4 = \mathrm{Nonlinear}(LL'_4) \tag{28}$$

Where $\mathrm{Nonlinear}(x) = x^2 + \mathrm{Exp}(x) \mod 2^b$, with Exp representing the exponential function.

In the last step, substitute values in LL''_4 using an S-box and take the inverse wavelet transform to generate the final encrypted image (E_{final}). An S-box is a substitution table used to obscure the relationship between plaintext and ciphertext. For this, we use a predefined S-box S to replace values in LL''_4 using Eq. 29. The algorithm for processing $T''_{i,j}$ is detailed in Algorithm 2.

$$E_{final} = S\left[\mathrm{mod}(LL''_4)\right] \tag{29}$$

function encrypt_image(T, wavelet_filters, A, S, b)
 $[LL_1, LH_1, HL_1, HH_1] \leftarrow$ DWT(T)
 $[LL_2, LH_2, HL_2, HH_2] \leftarrow$ DWT(LL_1)
 $[LL_3, LH_3, HL_3, HH_3] \leftarrow$ DWT(LL_2)
 $[LL_4, LH_4, HL_4, HH_4] \leftarrow$ DWT(LL_3)
 LL_4' $\leftarrow$ A $\cdot$ $LL_4 \cdot A^T$
 LL_4" $\leftarrow$ Nonlinear(LL_4')
 for i **in** range(0, size(LL_4)) **do**
  for j **in** range(0, size(LL_4)) **do**
   if mod($LL''_{4_{i,j}}$) < size(S) **then**
    $LL_4^{temp} \leftarrow$ S[mod($LL''_{4i,j}$)]
    if LL_4^{temp} **is not** NULL **then**
     $LL_{4_{i,j}}^{final} \leftarrow LL_4^{temp}$
    else
     $LL_{4_{i,j}}^{final} \leftarrow 0$
   else
    if $LL''_{4_{i,j}} > 128$ **then**
     $LL_{4_{i,j}}^{final} \leftarrow$ rotate(S[mod($LL''_{4_{i,j}}$)])
    else
     $LL_{4_{i,j}}^{final} \leftarrow$ S[mod($LL''_{4_{i,j}}$)]
   end for
  end for
 return LL_4^{final}

Algorithm 2. Encryption of image using DWT and S-box substitution

Figure 5 displays both the plaintext images and their corresponding ciphertext images produced by the proposed encryption framework. As shown in Fig. 5e, k, q, w, the plaintext information is entirely concealed in the ciphertext images, which demonstrates that the proposed framework effectively encrypts images of varying gray levels. This is especially noteworthy as images with fewer gray levels often exhibit high data correlation, and the proposed framework handles such correlated images efficiently. To decrypt the plaintext image from the ciphertext image, the decryption procedure can be followed as outlined below:

Decryption procedure

Decryption of the proposed encryption process involves reversing each of the operations applied during encryption process. A step-by-step explanation of the decryption process is as follows:

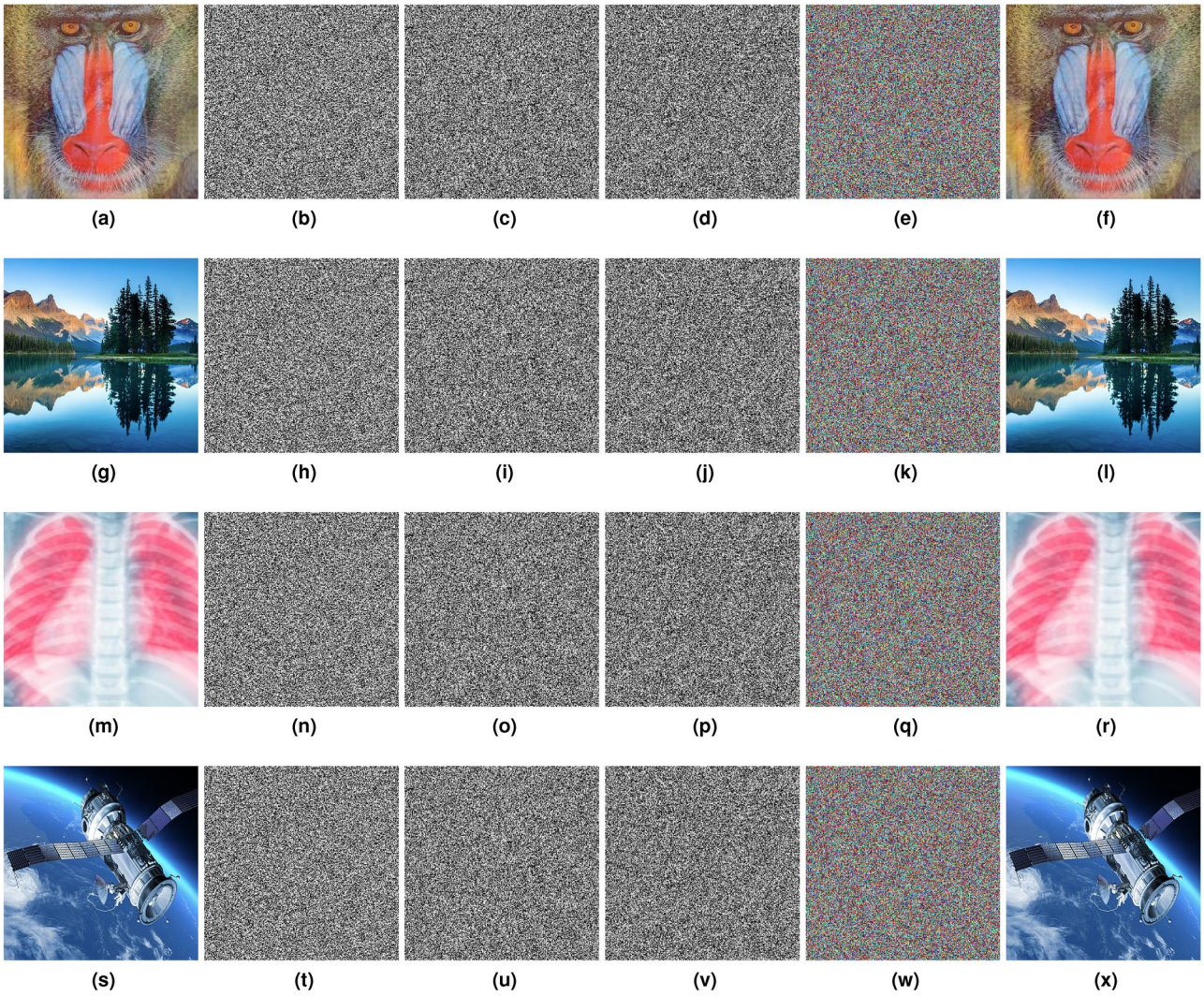


Fig. 5. (**a**, **g**, **m**, **s**) Original plaintext images, (**b**–**d**, **h**–**j**, **n**–**p**, **t**–**v**) their respective encrypted components, and (**e**, **k**, **q**, **w**) the corresponding fully encrypted color images, and (**f**, **l**, **r**, **x**) the decrypted versions.

- *Inverse of DWT (discrete wavelet transform)* First, the DWT applied during encryption is reversed using Eq. reveq. The inverse DWT (IDWT) reconstructs the image from the four subbands generated in the encryption phase.

$$T'' = \text{IDWT}(LL_1, LH_1, HL_1, HH_1) \tag{30}$$

- *Inverse diffusion operation* To reverse the diffusion operation, the inverse transformation is applied using the same key matrix K generated during encryption. In the decryption phase, the key matrix K is reconstructed from the same chaotic sequences as in the encryption process. The inverse diffusion equation is applied to the intermediate result from the previous step, as given in Eq. 31.

$$T'_{i,j} = \left(D''_{i,j} - j \cdot \left\lfloor \frac{K_{i,j}}{2} \right\rfloor\right) \mod 2^b \tag{31}$$

- *Inverse confusion operation* The confusion operation uses the combined chaotic sequences to obfuscate the original image matrix. To reverse this, the applied confusion matrix S and the nonlinear function $g(d_i, s_j)$ are subtracted. The inverse confusion operation is given by Eq. 32.

$$T_{i,j} = (T'_{i,j} - S_{i,j} - g(d_i, s_j)) \mod 2^b \tag{32}$$

Here, $g(d_i, s_j)$ is the same nonlinear function used during encryption.

- *Quantum-to-classical conversion* The final quantum state $|\Psi_f\rangle$ is measured, and the results are converted into grayscale pixel values. Since this step uses classical measurements, the quantum-to-classical conversion is applied directly on the measurements obtained from the quantum state $|\Psi_f\rangle$. The corresponding pixel value is then derived from the measurement outcomes.
- *Inverse of quantum substitution-permutation network (QSPN)* Reverse the substitution and permutation operations applied during encryption. Since the QSPN uses unitary matrices, the inverse of the matrices used during encryption is applied. Inverse substitution and permutation are performed as given in Eq. 33.

$$|\Psi'\rangle = P^{-1}\left(S^{-1}\left(\cdots P^{-1}\left(S^{-1}|\Psi\rangle\right)\cdots\right)\right) \tag{33}$$

- *Inverse unitry transformation and permutation* Each block's unitary transformation U_b is inverted. This operation reverses the scrambling of the quantum states that was performed in the encryption process. The inverse of the permutation matrix P is also applied to recover the order of the blocks.
- *Reconstruct the classical image* Finally, after reversing all quantum operations, the image matrix is reconstructed to its original form.

Figure 5f, l, r, x illustrates the decrypted images, which reveals that the proposed encryption framework can accurately recover the plaintext data, with all the original information clearly visible in the decrypted images.

Experimental result and analysis

The implementation of the proposed encryption framework was carried out using MATLAB 2020 on a system with the following hardware specifications: 8GB RAM, 512GB SSD, Windows 11, and an 11th Generation Intel Core i5 processor. All the images used in this research are obtained through google search from publicly available online repositories. Furthermore, these images are used for comparison with the existing encryption schemes, which are of the same size and share similar characteristics, including natural scenes, standard test images (e.g., Lena, Baboon, Peppers), medical images, face images, and satellite images. The evaluation involved a series of statistical analyses, including entropy analysis, key sensitivity analysis, key space analysis, and histogram analysis. Additionally, the resistance of the proposed framework to various cyberattacks, such as brute force, noise, and clipping attacks, is also tested. Furthermore, all the statistical results presented in this section represent the average of the encrypted R, G, and B components.

Histogram analysis

Histogram analysis reveals the distribution shape, central tendency, and spread of data which highlights the pixel frequency of an image[55]. For robust encryption, the ciphertext image's histogram should differ from that of the plaintext image, and the pixel distribution should be nearly uniform.

Different images are tested for the evaluation of the proposed encryption framework. The histogram analysis reveals that the histograms of the ciphertext images are nearly flat and significantly different from the histograms of the plaintext images. This contrast indicates that the encryption framework effectively disrupts the original image distribution, demonstrating its ability to obscure the image content and enhance security.

Figure 6 presents the histogram analysis for the proposed encryption framework. Figure 6b–d, j–l illustrate the histograms of the plaintext R,G,B components, whereas Fig. 6f–h, n–p depict the histograms of the encrypted R,G,B components. The nearly flat histograms of the encrypted R,G,B components indicate that the proposed encryption scheme is resistant to histogram attacks.

Histogram variance analysis

Histogram variance analysis in image encryption involves comparing the variance of pixel value distributions in plaintext and ciphertext images to assess encryption effectiveness[56,57]. To compute the histogram variance, first, compute the histograms for both images, which show the frequency of each pixel value. Next, calculate the mean and variance for these histograms. Mathematically, the histogram variance can be calculated using Eq. 34.

$$\text{Variance}_{\text{cipher}} = \frac{\sum_{i=0}^{255}(x_i - \text{Mean}_{\text{cipher}})^2 \cdot f_i}{\sum_{i=0}^{255} f_i} \tag{34}$$

where x_i is the pixel value at index i ranges $i \in [0, 55]$ for a grayscale image. The mean ($\text{Mean}_{\text{cipher}}$) is $\text{Mean}_{\text{cipher}} = \frac{\sum_{i=0}^{255}(x_i \cdot f_i)}{\sum_{i=0}^{255} f_i}$. f_i represents the frequency of pixel value x_i.

For strong encryption, the ciphertext histogram should exhibit a lower variance and a more uniform distribution compared to the plaintext histogram. This indicates that the encryption scheme effectively conceal the pixel values and prevents recognizable patterns. In Table 2, variance values for various proposed and existing methods are displayed which shows that the histogram variance for the proposed encryption framework is comparatively lower than that of existing methods. This indicates that the proposed framework is more effective in terms of histogram variance analysis.

Lossless analysis

Lossless analysis is performed to assess the extent of information loss when recovering plaintext data from ciphertext[60,61]. This is statistically evaluated using peak signal-to-noise ratio (PSNR)[62] and mean squared error (MSE)[63]. Mathematically, PSNR and MSE are computed using Eqs. 35 and 36, respectively.

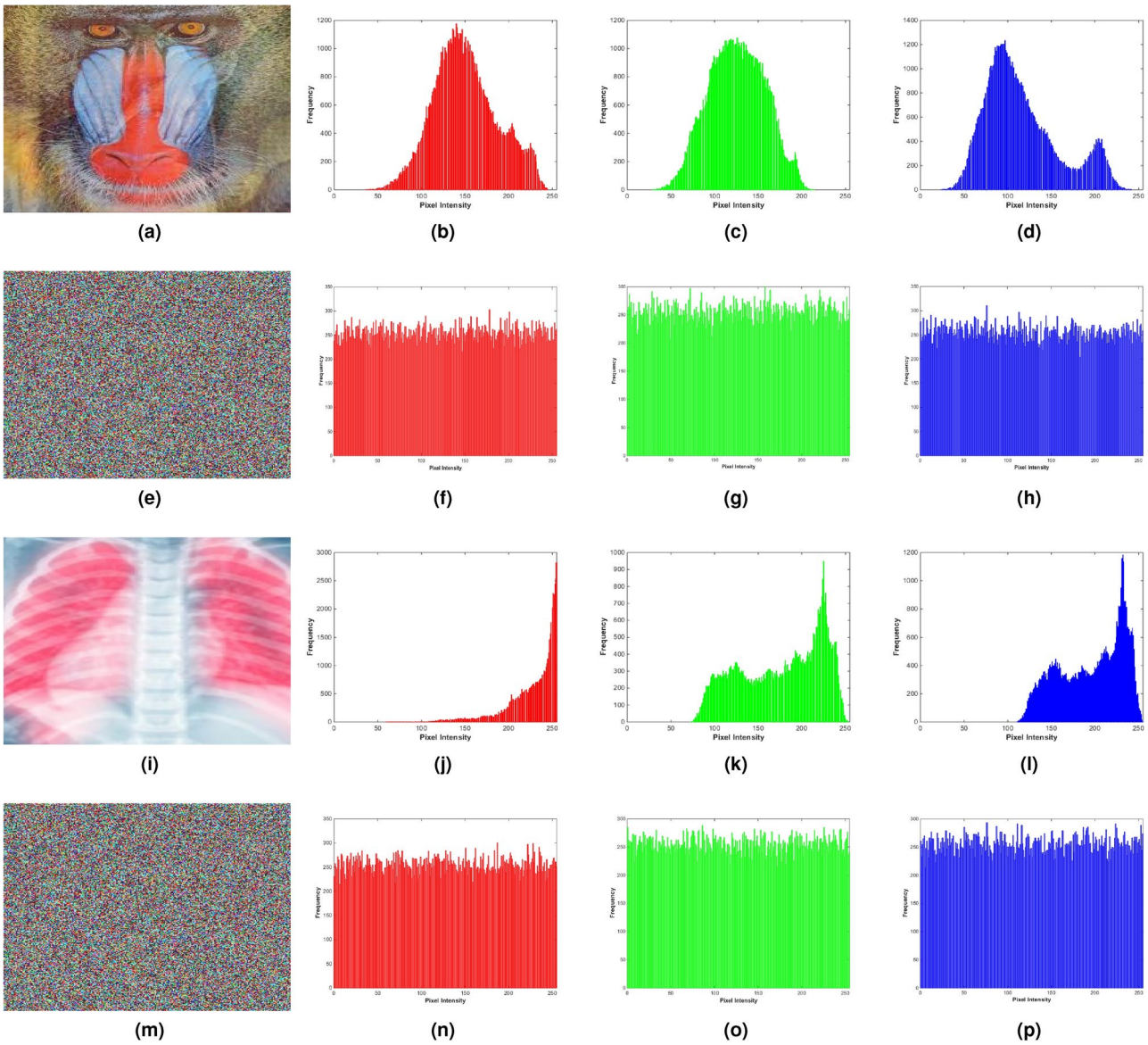


Fig. 6. (**a**, **f**, **k**, **p**) Original plaintext images, (**b**–**d**, **g**–**i**, **l**–**n**, **q**-**s**) their respective encrypted components, and (**e**, **j**, **o**, **t**) the corresponding fully encrypted color images.

Image size	Quantum	Xray	Baboon	Tumor	Average
$256 \times 256 \times 3$	259.35	256.21	259.48	257.65	258.17
	256.12	255.99	257.66	256.99	256.69
$256 \times 256 \times 3$	255.61	256.15	257.66	259.45	257.21
	254.36	255.02	256.07	257.64	255.77
$512 \times 512 \times 3$	256.37	256.34	255.33	259.45	256.87
	255.17	256.98	257.15	254.37	255.91
Comparison with existing schemes					
Ref.[22]	260.35	261.21	258.64	260.69	260.30
Ref.[58]	260.15	261.33	261.99	260.33	260.44
Ref.[59]	259.60	260.45	261.03	259.83	260.00

Table 2. Statistics of histogram variance.

Image size	Metrics	Quantum	Xray	Baboon	Tumor	Average
$256 \times 256 \times 3$	PSNR	∞	∞	∞	∞	∞
	MSE	0	0	0	0	0
$256 \times 256 \times 3$	PSNR	∞	∞	∞	∞	∞
	MSE	0	0	0	0	0
$512 \times 512 \times 3$	PSNR	∞	∞	∞	∞	∞
	MSE	0	0	0	0	0
$512 \times 512 \times 3$	PSNR	∞	∞	∞	∞	∞
	MSE	0	0	0	0	0
Comparison with existing schemes						
Ref.[22]	PSNR	40.65	41.16	42.77	44.36	42.23
	MSE	10.16	9.46	8.65	7.16	8.85
Ref.[58]	PSNR	43.31	45.16	45.99	49.01	45.86
	MSE	8.12	6.47	6.38	5.94	6.72
Ref.[59]	PSNR	46.13	41.01	43.6	39.99	42.68
	MSE	5.79	9.31	8.32	11.89	8.82

Table 3. Lossless analysis.

Image size	Images	Ref.[58]	Ref.[59]	Ref.[22]	Ref.[23]	Ref.[64]	Proposed
$256 \times 256 \times 3$	Quantum	7.9876	7.9960	7.9936	7.9942	7.9986	7.9992
	Xray	7.9777	7.9971	7.9936	7.9989	7.9990	7.9993
$512 \times 512 \times 3$	Baboon	7.9789	7.9982	7.9936	7.9951	7.9973	7.9991
	Tumor	7.9888	7.9940	7.9946	7.9973	7.9987	7.9990

Table 4. Entropy analysis.

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{L^2}{\text{MSE}}\right) \tag{35}$$

$$\text{MSE} = \frac{1}{N \cdot M}\sum_{i=1}^{N}\sum_{j=1}^{M}[I(i,j) - K(i,j)]^2 \tag{36}$$

Here, the plaintext and ciphertext images both have dimensions $N \times M$. The pixel values $I(i,j)$ and $K(i,j)$ correspond to the positions (i,j) in the plaintext and ciphertext images, respectively. The term L denotes the number of gray levels in the images.

Any MSE value greater than zero indicates that the decrypted image differs from the original plaintext image, even if their overall content appears visually similar. Table 3 presents the PSNR and MSE values, demonstrating the lossless nature of the proposed and existing encryption algorithms. It shows that the MSE values for the proposed encryption framework are zero across all images, implying an infinite PSNR according to Eq. 35, since PSNR and MSE are inversely proportional. In contrast, the existing encryption algorithms show non-zero MSE values. Although these values are not excessively high which indicates that the decrypted images are visually close to the originals, the exact pixel values differ, resulting in lower PSNR values for the existing methods.

Entropy analysis

Entropy measures the level of randomness in the final ciphertext image. For robust encryption, high randomness in the ciphertext is essential. In the context of 8-bit grayscale images, the ideal entropy value is 8. Mathematically, it can be calculated using Eq. 37.

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) \tag{37}$$

where: $p(t_i)$ is the probability of occurrence of the t-th pixel value.

In this research, 8-bit grayscale images are encrypted, and Table 4 presents the entropy values for both the proposed and existing encryption schemes. It can be observed that the entropy values for the proposed method are close to 8. Specifically, the average entropy value for the proposed scheme is approximately 7.9998, whereas for existing schemes, it is about 7.9989. This indicates that the proposed encryption framework offers slightly better resistance to entropy attacks compared to the existing encryption schemes.

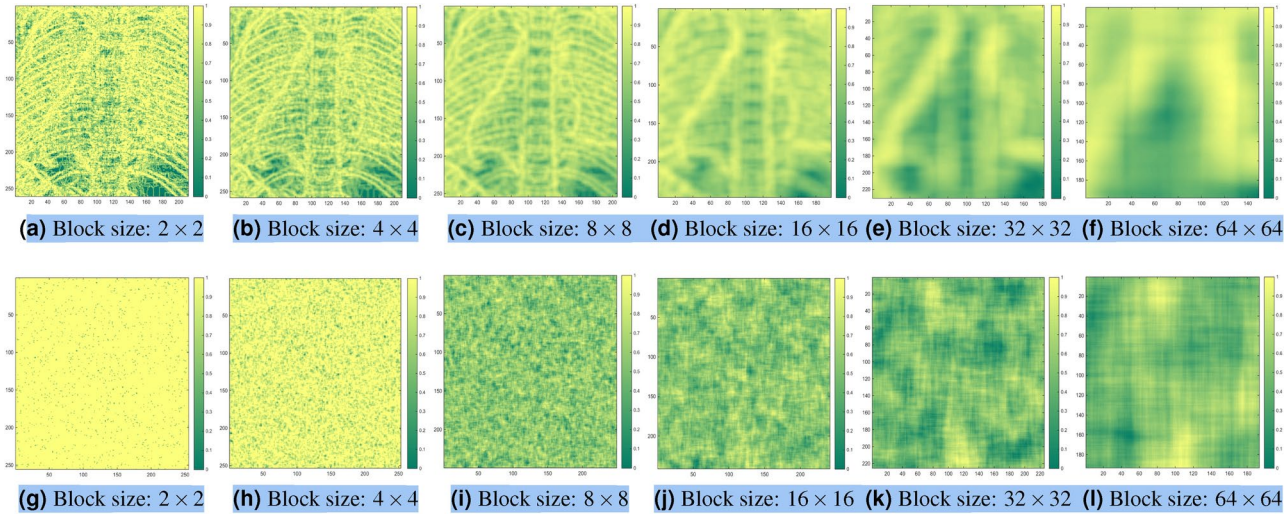


Fig. 7. Local entropy analysis for varying block sizes.

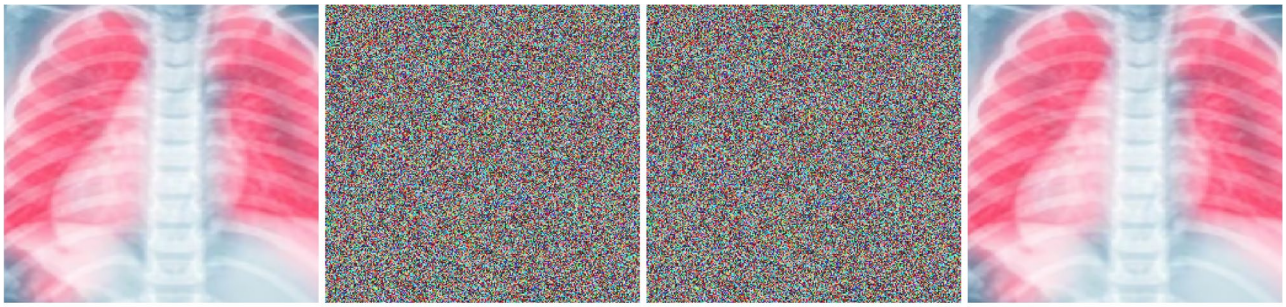


Fig. 8. Key sensitivity analysis.

Local entropy

Local entropy analysis evaluates the randomness of an encrypted image within different-sized regions. In this research, the plaintext and their corresponding encrypted images are divided into six different block sizes: 2×2, 4×4, 8×8, 16×16, 32×32, and 64×64. The entropy is computed for each region using the standard entropy formula as give in Eq. 37. The results show that for smaller block sizes (2×2, 4×4, 8×8), entropy variations are more prominent. This is because smaller blocks capture localized details and randomness better. As the block size increases (16×16, 32×32, 64×64), entropy values become smoother. This indicates a more uniform encryption effect at broader scales.

A graphical illustration of the local entropy analysis for the plaintext image and its corresponding encrypted image, evaluated across different block sizes, is presented in Fig. 7. Specifically, Fig. 7a–f depict the local entropy plots for the plaintext image at various block sizes, while Fig. 7g–l display the local entropy plots for the encrypted image at the same block sizes. Notably, even for the smallest block size (2×2), the proposed encryption framework effectively encrypts all plaintext information, demonstrating the robustness and effectiveness of the encryption approach.

Key sensitivity analysis

To enhance the security of encrypted data, the sensitivity of the secret keys used in the encryption framework is crucial. Sensitivity here means that even a minor change in the secret keys will result in decryption failure. In the proposed encryption method, there are nineteen secret keys, namely $\xi, \phi, d_1, d_2, \cdots, d_7, n, p, x, u_1, u_2, k, L, o, m$, and v.

To assess the sensitivity of each key, a small perturbation, $\Delta = 10^{-15}$, is added to each secret key, creating new modified keys: $\xi' = \xi + 10^{-15}$, $\phi' = \phi + 10^{-15}$, $d_1' = d_1 + 10^{-15}$, $d_2' = d_2 + 10^{-15}$, and so on for each key. These modified keys are then used to decrypt the ciphertext image.

The decrypted images obtained with these modified keys are shown in Fig. 8. It is evident from the figure that the modified keys fail to recover the original plaintext image, as the decrypted images contain no meaningful information about the plaintext.

Keyspace analysis

Key space analysis is employed to assess an encryption algorithm's resilience against brute-force attacks. In a brute-force attack, an attacker systematically tries every possible combination of secret keys to find the correct one. Thus, to effectively resist such attacks, the encryption algorithm must use secret keys of sufficient size. In the proposed work, a total of nineteen secret keys are utilized, each with a sensitivity of 10^{-15} as detailed in Section Key sensitivity analysis. This implies that each key has a size of 10^{15}. The total key space for the proposed encryption scheme is calculated using Eq. 38.

$$\begin{cases} 10^{19\times 15} = 10^{285} \\ 10^{285} \approx (2^{3.32193})^{285}, \quad (2^{3.32193})^{285} = 2^{3.32193\times 285} \\ 3.32193 \times 285 \approx 947.862, \quad 10^{285} \approx 2^{947.862} \end{cases} \tag{38}$$

According to Alvarez's criteria[65], an encryption scheme must have a key space of at least 2^{100} to be considered sufficiently resistant to brute-force attacks. The proposed encryption scheme meets this requirement, thereby satisfying Alvarez's key space criteria and ensuring robustness against brute-force attacks.

Computational complexity analysis

In addition to statistical security analysis, evaluating the encryption algorithm's computational complexity is essential. For real-time applications, the encryption algorithm must be time-efficient to ensure it can be effectively utilized. The proposed encryption algorithm is tested with two different image sizes: 256×256 and 512×512 . To measure the time taken by the proposed encryption framework for its complete encryption process, the MATLAB "tic-toc" command was used. Since the "tic-toc" command yields slightly different times with each run, the encryption process is executed five times to compute an average time.

Table 5 presents the computational time values for the proposed encryption algorithm compared to existing algorithms. To ensure a fair comparison, the existing encryption algorithms are implemented on the same platform as the proposed algorithm. According to Table 5, the proposed encryption scheme generally outperforms the existing ones in terms of computational efficiency, except for the encryption scheme presented in[58], which is marginally faster.

Correlation analysis

Correlation analysis examines the relationship between adjacent pixels based on their similarity. The closer the pixel values are to each other, the higher the correlation between them. Typically, plaintext images exhibit high pixel correlation, reflecting a significant amount of meaningful information. To ensure effective encryption, the goal is to minimize pixel correlation in the encrypted images. To assess the effectiveness of the proposed encryption scheme in terms of pixel correlation, a random sample of 10,000 pixels is taken from both the original images and their corresponding encrypted versions. The correlation between the image pixels is calculated in all directions-horizontal, vertical, and diagonal by extracting a random sample of 10,000 pixels from both the original images and their corresponding encrypted versions. Mathematically, pixel correlation can be determined using Eq. 39.

$$\begin{cases} \mathrm{Corr}(P,T) = \frac{\mathrm{Cov}(P,T)}{\sigma_P \sigma_T} \\ \mathrm{Cov}(P,T) = \frac{1}{N}\sum_{i=1}^{N}(p_i - \bar{p})(t_i - \bar{t}) \\ \bar{p} = \frac{1}{N}\sum_{i=1}^{N} p_i \\ \bar{t} = \frac{1}{N}\sum_{i=1}^{N} t_i \\ \sigma_P = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(p_i - \bar{xp})^2} \\ \sigma_T = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_i - \bar{t})^2} \end{cases} \tag{39}$$

Where: $\mathrm{Cov}(P,T)$ is the covariance between p and t, σ_p is the standard deviation of p, σ_t is the standard deviation of t. Table 6 presents the correlation values for both the proposed and existing encryption schemes. It is evident that the correlation values for the proposed encryption scheme are lower compared to the existing ones.

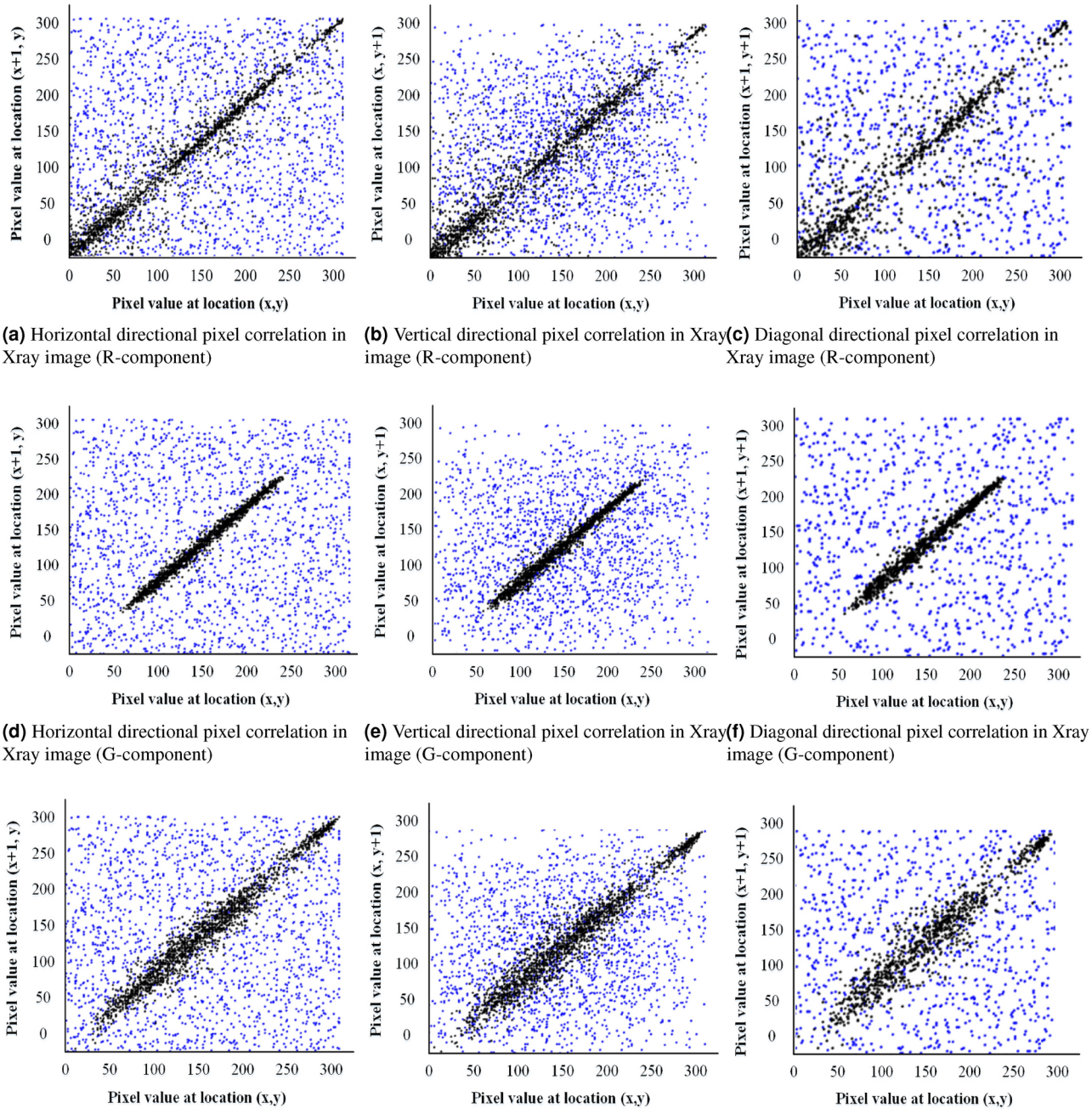
In addition to statistical correlation analysis, visual analysis is also performed. For the visual correlation analysis, scatter plots of the pixel values for the plaintext and ciphertext images are shown in Fig. 9. In these plots, black dots represent the pixel correlations of the plaintext image, while blue dots denote the pixel correlations of the ciphertext image. A higher density of dots indicates a greater pixel correlation. The scattered arrangement of blue dots shown in Fig. 9 illustrates a reduced correlation between the ciphertext image pixels.

Image size	Images	Ref.[58]	Ref.[59]	Ref.[22]	Ref.[23]	Ref.[64]	Proposed
$256 \times 256 \times 3$	Quantum	0.013	0.214	0.351	0.039	0.79	0.002
	Xray	0.015	0.263	0.045	0.043	0.94	0.001
$512 \times 512 \times 3$	Baboon	0.035	0.053	0.086	0.088	1.56	0.004
	Tumor	0.040	0.060	0.090	0.096	1.63	0.003

Table 5. Computational time analysis.

Size	Images	Ref.[58]	Ref.[59]	Ref.[22]	Ref.[23]	Ref.[64]	Proposed
$256 \times 256 \times 3$	Quantum	0.0026	0.0016	0.0017	0.0025	0.0018	0.0001
	Xray	0.0016	0.0014	-0.0025	-0.0019	0.0010	-0.0015
$512 \times 512 \times 3$	Baboon	0.0022	-0.0020	-0.0012	0.0020	-0.0014	-0.0001
	Tumor	0.0027	-0.0019	-0.0016	-0.0013	0.0031	-0.0015

Table 6. Correlation analysis.



(a) Horizontal directional pixel correlation in Xray image (R-component)

(b) Vertical directional pixel correlation in Xray image (R-component)

(c) Diagonal directional pixel correlation in Xray image (R-component)

(d) Horizontal directional pixel correlation in Xray image (G-component)

(e) Vertical directional pixel correlation in Xray image (G-component)

(f) Diagonal directional pixel correlation in Xray image (G-component)

(g) Horizontal directional pixel correlation in Xray image (B-component)

(h) Vertical directional pixel correlation in Xray image (B-component)

(i) Diagnoal directional pixel correlation in Xray image (B-component)

Fig. 9. Correlation analysis of Xray images in horizontal, vertical, and diagonal directions.

Noise and clipping attack

To cause decryption failure at the receiver end, an eavesdropper often attempts noise and cropping attacks. In a noise attack, the eavesdropper introduces random noise into the ciphertext image to disrupt decryption. To assess the resilience of the proposed encryption scheme against such noise attacks, salt and pepper noise is added to the ciphertext image according to Eqs. 40–43, affecting 10% of the image pixels.

$$M_{noise}(i,j) = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{with probability } (1-p) \end{cases} \tag{40}$$

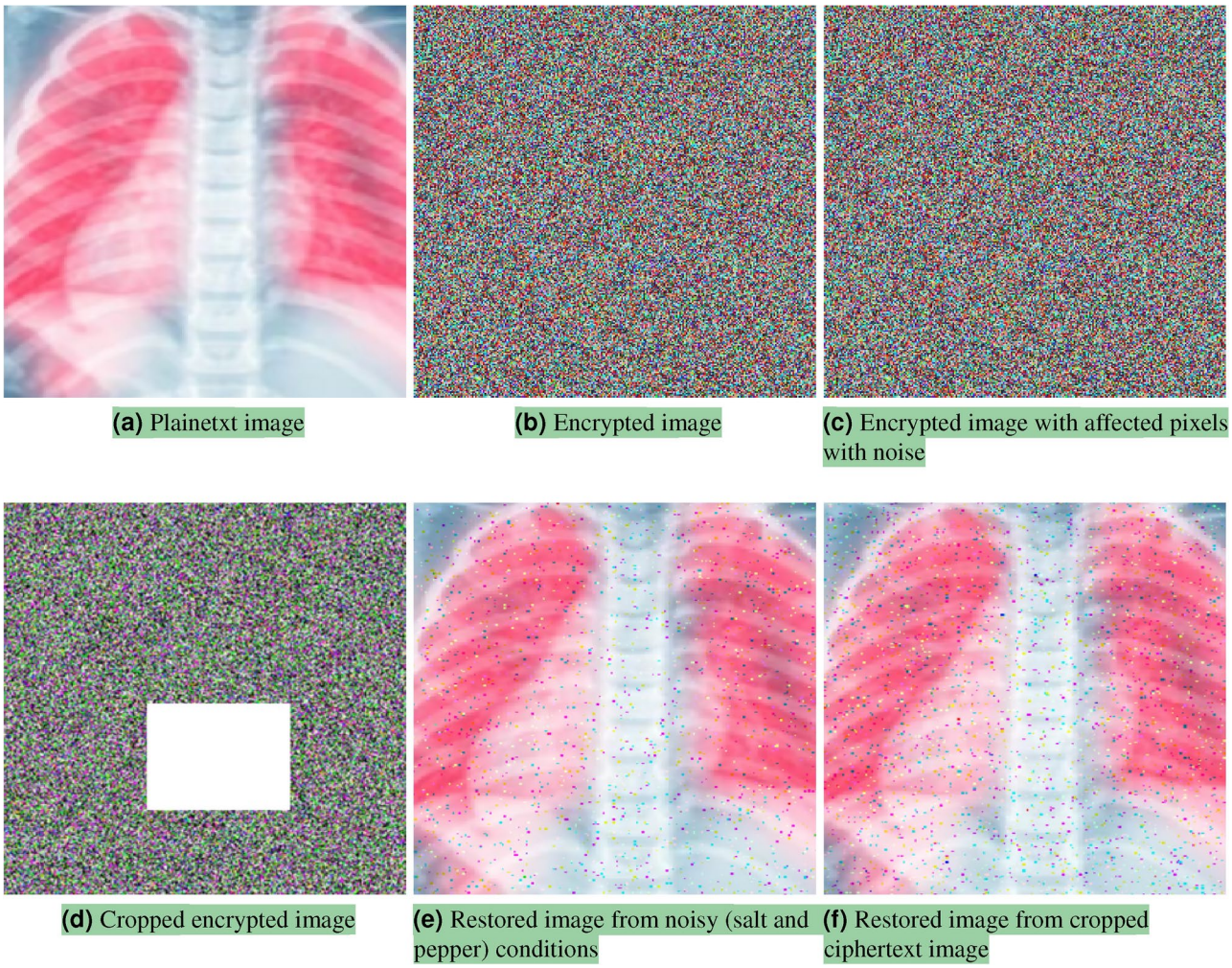
$$M_{salt}(i,j) = \begin{cases} 1 & \text{with probability } \frac{p}{2} \\ 0 & \text{with probability } 1-\frac{p}{2} \end{cases} \tag{41}$$

$$M_{pepper}(i,j) = \begin{cases} 1 & \text{with probability } \frac{p}{2} \\ 0 & \text{with probability } 1-\frac{p}{2} \end{cases} \tag{42}$$

$$I_{noisy}(i,j) = \begin{cases} 255 & \text{if } M_{salt}(i,j) = 1 \text{ and } M_{noise}(i,j) = 1 \\ 0 & \text{if } M_{pepper}(i,j) = 1 \text{ and } M_{noise}(i,j) = 1 \\ I(i,j) & \text{otherwise} \end{cases} \tag{43}$$

where $p = 0.1$ (10% noise) is the total probability of noise. $C(i,j)$ is the ciphertext pixel value. $C_{noisy}(i,j)$ is the pixel value in the noisy image. Ensure that $M_{salt} + M_{pepper} = M_{noise}$. After adding salt and pepper noise, the plaintext image is recovered as depicted in Fig. 10. Figure 10a–d shows the plaintext, encrypted, encrypted noisy, and encrypted cropped images, respectively. It is evident from Fig. 10e that the plaintext information remains visible, with only minor pixel distortion.

Furthermore, to assess the robustness of the proposed encryption framework against cropping attacks, a portion of the ciphertext image is cropped, and the plaintext image is then decrypted from the cropped



(a) Plainetxt image **(b)** Encrypted image **(c)** Encrypted image with affected pixels with noise

(d) Cropped encrypted image **(e)** Restored image from noisy (salt and pepper) conditions **(f)** Restored image from cropped ciphertext image

Fig. 10. Noise and cropping attack analysis.

Image size	10×10	50×50	100×100	200×200	300×300	400×400	500×500
Memory complexity ($O(n^2)$)	O(100)	O(2500)	O(10,000)	O(40,000)	O(90,000)	O(160,000)	O(250,000)
Memory usage (in MB)	0.10	0.25	1	4	9	16	25

Table 7. Memory complexity for different image sizes.

Image size	10×10	50×50	100×100	200×200	300×300	400×400	500×500
Energy consumption ($O(n^2 \log n)$)	O(100 log 10)	O(2500 log 50)	O(10000 log 100)	O(40000 log 200)	O(90000 log 300)	O(160000 log 400)	O(250000 log 500)
Estimated energy (in Joules)	0.01 J	0.2 J	1.0 J	5.0 J	12.0 J	20.0 J	35.0 J

Table 8. Estimated energy consumption for different image sizes.

ciphertext. Figure 10f demonstrates that the encryption framework effectively resists cropping attacks, as the plaintext information is recovered with only minor noise in the resulting image.

Memory complexity analysis

The memory complexity of the proposed encryption framework is analyzed by examining how the memory usage scales with increasing input image size. The memory complexity is modeled as $O(n^2)$, where n is the size of the input image $(n \times n)$. The memory complexity is modeled as $O(n^2)$, where n is the size of one side of the image (i.e., the image is $n \times n$). This implies that the amount of memory required to process the image grows quadratically with respect to the image size. For instance, if $n = 10$ (a 10×10 image), the complexity would be $O(10^2) = O(100)$. Similarly, for $n = 500$ (a 500×500 image), it would be $O(500^2) = O(250,000)$. Thus, for an image of size $n \times n$, the memory complexity M_c can be mathematically expressed as:$M_c = O(n^2$. The experiments are conducted with images of sizes ranging from 10×10 to 500×500, recording the memory complexity for each input size. The results are summarized in Table 7.

From Table 7, it can be seen that memory usage grows quadratically with the image size, as expected from the $O(n^2)$ complexity model. As the image size increases, memory usage increases proportionally. For instance, a 100×100 image requires approximately 1MB of memory, whereas a 500×500 image demands 25MB of memory. This growth pattern is consistent with the theoretical model, and the algorithm can handle images up to 500×500 with acceptable memory usage.

For IoT devices with limited memory, images up to 100×100 (1MB) are ideal for encryption. However, larger images can be managed using techniques like compression, dynamic resizing, partitioning into smaller blocks, and memory management strategies such as swapping and streaming. These methods reduce memory use and allow for the efficient encryption of larger images. While larger images consume more energy, the proposed encryption framework is designed to handle this by working effectively with IoT devices, which can manage energy demands through optimized processors or energy harvesting. This makes the framework ideal for IoT devices, enabling efficient encryption even for larger images.

Energy consumption estimation

The energy consumption is estimated using the model $O(n^2 \log n)$, where n represents the size of the input image. This model takes into account the computational cost, including matrix operations and transformations applied during encryption. Simulations are performed for different image sizes, ranging from 10×100 to 500×500, and calculated the estimated energy consumption for each size. The memory usage M_u is calculates as: $M_u = n^2 \times \text{Memory per pixel}$. Where n^2 is the total number of pixels in an $n \times n$ image. Memory per pixel depends on the image type (for example, 1 byte per pixel for grayscale, or 3 bytes per pixel for RGB images). For a grayscale image (1 byte per pixel), for an image size of 100×100: $M_u = 100^2 \times 1 \text{ byte} = 10,000 \text{ bytes} = 10 \text{ KB}$. For a color image (3 bytes per pixel), for the same 100×100 image:$M_u = 100^2 \times 3 \text{ bytes} = 30,000 \text{ bytes} = 30 \text{ KB}$.

The results of the energy consumption analysis are shown in Table 8 where it can be seen that the energy consumption increases both quadratically and logarithmically with the image size. For example, a 100×100 image requires approximately 1.0J of energy, while a 500×500 image demands around 35.0J. This suggests that the proposed encryption framework, while efficient, will consume more power as the image size increases. This scaling is important to consider for resource-constrained IoT devices, where energy consumption must be minimized.

Key management

The proposed encryption scheme employs a robust key management system by integrating chaotic and quantum operations, and a key matrix to ensure high security and resistance against attacks. A key matrix, generated using chaotic map functions which plays a critical role in the diffusion process. This shows strong key sensitivity, meaning even a slight modification in the key leads to a significantly altered encrypted output. Moreover, the encryption achieves a vast key space of $2^{947.862}$ as mentioned in Section Keyspace analysis, making brute-force attacks computationally infeasible, while an entropy value of 7.9998 indicates near-perfect randomness. The security of the key management system is further validated through key sensitivity analysis as provided in Section Key sensitivity analysis, which confirms that unauthorized access is impossible without the exact key. By

leveraging chaotic maps, quantum encryption, and metaheuristic optimization, the proposed method provides a highly secure and efficient key management mechanism suitable for resource-constrained IoT environments.

Suitability of the 7D hyperchaotic system for the proposed encryption scheme

The 7D hyperchaotic system is evaluated to determine its suitability for the proposed encryption scheme using Lyapunov Exponent (LE), entropy-based loss function, key sensitivity loss function, and integrated loss function, as shown in Fig. 11. The Lyapunov Exponent plot (Fig. 11a) presents values between 0.88 and 0.9157, which indicates strong chaotic behaviour, ensures unpredictability, and enhances encryption security. The entropy-based loss function (Fig. 11b) exhibits entropy values close to 8, confirming a highly uniform distribution of pixel values in encrypted images, thereby preventing statistical attacks. The Key sensitivity-based loss function (Fig. 11c) demonstrates that even a minor key change (10^{-6}) leads to nearly 100% pixel variation, validating a strong avalanche effect, which is crucial for encryption robustness. The integrated loss function (Fig. 11d), evaluated across 50 test cases, combines these metrics, showing consistent performance in achieving high randomness, sensitivity, and security. These findings show that the 7D hyperchaotic system provides complexity and unpredictability, which makes it suitable for the proposed encryption framework.

Discussion and critical insights

This section explores the benefits of the proposed encryption framework for resource-limited devices, justifies software-based implementation over FPGA, and highlights its superiority in key space, entropy, efficiency, and attack resistance.

Benefits of quantum encryption in IoT

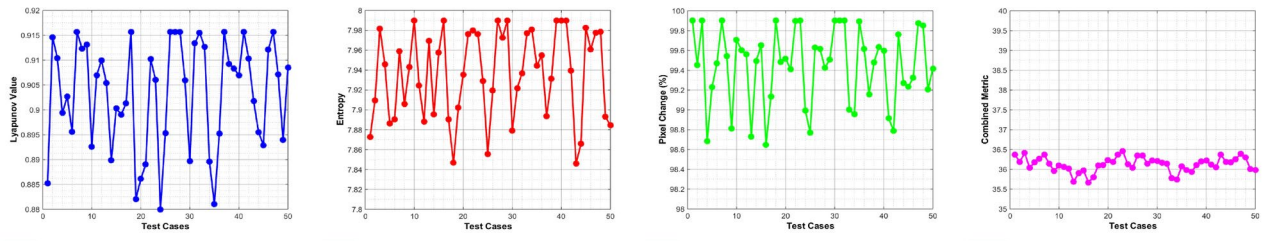
The adoption of quantum encryption in IoT security is motivated by the need for enhanced security without incurring excessive computational costs[66]. Traditional encryption algorithms rely on complex mathematical operations, which are inefficient for resource-constrained devices[67,68]. Quantum encryption leverages the principles of superposition[69] and entanglement[70], which enable more secure and lightweight encryption processes.

Why quantum operations in the proposed research

- *Enhanced key space expansion* Quantum-based key generation produces highly unpredictable keys due to quantum randomness. This significantly increases key space size which makes the brute-force attacks infeasible. The key space analysis of the secret keys utilized in the proposed encryption scheme is detailed in Section Keyspace analysis.
- *Reduced computational complexity* Classical encryption methods require extensive modular arithmetic, while quantum operations enable efficient transformations with fewer computational steps. This results in reduced processing power consumption which makes the proposed encryption process well-suited for IoT devices, as discussed in Section Computational complexity analysis.
- *Improved security metrics* Quantum encryption also strengthens key sensitivity and entropy. This makes the encryption framework resistant to statistical attacks. High entropy values, as discussed in Section Entropy analysis, ensure that the data encrypted with the proposed framework is uniformly random, thereby minimizing patterns that could be exploited by attackers.
- *Lower energy consumption* Quantum transformations operate in parallel states, requiring fewer iterations for encryption. As explained in Section Energy consumption estimation, the energy consumption estimation analysis demonstrates that the proposed encryption framework uses less than 20.0J of energy to encrypt an image of size $400 \times 400 \times 3$. Additionally, the comparison table of quantum encryption versus classical encryption for IoT is provided in Table 9, where it can be observed that the proposed encryption framework outperforms other well-known existing encryption frameworks.

Considered security attacks

The proposed encryption scheme is designed to resist multiple security attacks, ensuring robust protection for resource-constrained IoT devices. It effectively defends against brute-force attacks by utilizing a vast key space of $2^{947.862}$, making exhaustive key searches infeasible. The scheme also demonstrates resilience against



(a) yapunov Exponent ensures high unpredictability

(b) Entropy-Based Loss prevents statistical attacks

(c) Key Sensitivity validates the avalanche effect

(d) Integrated Loss confirms robustness across 100 test cases

Fig. 11. Performance evaluation of the 7D hyperchaotic system for encryption.

Metric	Classical encryption (AES, RSA)	Proposed quantum-based encryption
Key space size	2^{256} (AES-256)	$2^{947.862}$ (Proposed)
Entropy	≈ 7.95	7.9998 (Near Ideal)
Correlation coefficient	≈ 0.02	0.0001 (Lower is better)
Computational complexity	$O(n^2)$ (AES, RSA)	$O(n)$ (Quantum optimized)
Energy consumption	High due to modular arithmetic	Lower due to quantum parallelism
Attack resistance	Vulnerable to brute force, statistical, and side-channel attacks	Resistant due to quantum randomness

Table 9. Comparison of classical encryption vs. proposed quantum-based encryption.

noise attacks, where salt-and-pepper noise is introduced into the ciphertext, yet the plaintext image remains recoverable with minimal distortion. Similarly, it withstands cropping attacks, successfully reconstructing significant portions of the original image even when parts of the ciphertext are removed. Additionally, the encryption framework mitigates statistical attacks by achieving a near-uniform histogram distribution and high entropy values (7.9998), reducing the likelihood of frequency-based analysis. Moreover, correlation attacks are countered by significantly lowering adjacent pixel correlations in encrypted images (≈ 0.0001), making it difficult for attackers to detect patterns. These security measures, combined with its lightweight design, make the proposed encryption scheme a strong candidate for securing IoT communications against various cyber threats.

Justification for not using FPGA implementation

In this research, the proposed encryption algorithm is implemented and evaluated using MATLAB, which focuses on algorithmic efficiency, security strength, and computational feasibility for resource-constrained IoT devices. While FPGA-based implementation is often used for hardware validation, it requires dedicated hardware design, synthesis, and optimization, which is beyond the scope of this research. The primary objective of our research is to establish the theoretical feasibility of the proposed encryption scheme and evaluate its performance through rigorous statistical and computational complexity analyses.

Implementing the algorithm in MATLAB allows for rapid prototyping, in-depth statistical evaluations, and comprehensive performance benchmarking against established encryption techniques. Also, MATLAB provides an effective platform for validating security parameters such as entropy, correlation, keyspace, and robustness against attacks. Furthermore, resource-constrained IoT devices vary in hardware configurations, which makes a single FPGA implementation insufficient to generalise results across diverse IoT architectures. Instead of implementing the algorithm on an FPGA, a detailed complexity analysis, including computational and memory complexity, is conducted to justify the suitability of the proposed encryption framework for IoT applications. Moreover, this study provides MATLAB-based implementation with a controlled and replicable environment for assessing the core contributions of our encryption framework.

Significance and impact

The proposed work differs from and surpasses the encryption scheme proposed in[71] by integrating metaheuristic optimisation for key generation, enhancing randomness and resilience against brute-force attacks, unlike traditional chaotic key-based encryption. Optimised for resource-constrained IoT devices, the proposed approach reduces computational overhead while achieving high entropy (7.9998), low correlation (0.0001), and an expansive key space ($2^{947.862}$), outperforming conventional and other existing encryption schemes[22,23,58,59,64,71]. Unlike the encryption scheme proposed in[71], DWT-based substitution is incorporated, leveraging multi-resolution analysis to further strengthen security. Additionally, the framework demonstrates superior resistance against noise, cropping, and differential attacks. This ensures robustness under real-world adversarial conditions.

The significance and the impact of the proposed encryption framework are demonstrated through its superior security performance and computational efficiency as outlined in Section Experimental result and analysis. The significantly improved security and computational efficiency make it highly suitable for real-time IoT applications. Extensive testing shows its robustness against cropping, noise, and differential attacks. This ensures the data integrity even under adversarial conditions. Unlike existing approaches, which often neglect IoT-specific constraints[22,23,25], the proposed encryption framework is computationally lightweight, minimising energy consumption while maximising security. By bridging quantum security principles with classical cryptographic techniques, this work also presents a future-proof, scalable, and attack-resilient encryption model. Furthermore, the proposed framework provides a promising foundation for integration into future IoT systems, where security, low power consumption, and real-time processing are essential for practical applications. This work also opens avenues for further optimization, particularly in hardware-based implementations like FPGA or ASIC, where the balance between performance and energy efficiency is critical.

Moreover, to substantiate the claim of being lightweight for Wireless Sensor Networks, extensive testing and evaluation were conducted, as detailed in Section Experimental result and analysis. In our experiments, the proposed encryption algorithm is evaluated on WSN device such as Xbee Module with constrained processing capabilities. The results demonstrated that the algorithm is capable of encrypting data in under one second for images of sizes 256x256 and 512x512, as presented in Section Computational complexity analysis. This demonstrates its fast execution, which is crucial for real-time WSN applications. Additionally, the computational overhead, including processor utilization and memory usage, is minimal, as outlined in Section Memory

complexity analysis, ensuring that the encryption scheme does not unduly burden the resource-constrained devices in the network. Furthermore, energy consumption tests showed that the algorithm's low computational demands result in lower power usage compared to more complex encryption methods, which is crucial for battery-powered sensor nodes in WSNs, as detailed in Section Energy consumption estimation. These empirical results validated the lightweight nature of our proposed method, showing the suitablebili of the proposed encryutpion framework for resource-constrained environments like WSNs.

Conclusion

In this research a lightweight encryption algorithm tailored for resource-constrained IoT devices is proposed. The proposed technique integrates multiple techniques for encryption, such as quantum encryption techniques, chaotic systems, and the discrete wavelet transform (DWT). The proposed approach addresses the need for both efficiency and security in IoT environments, where traditional encryption methods often fall short due to resource limitations. By leveraging advanced techniques, the proposed encryption parameters are effectively fine-tuned to enhance performance without compromising security. The unique combination of quantum encryption, chaotic maps, and confusion-diffusion operations provides a robust framework for data protection, while DWT contributes to making the proposed encryption framework more time-efficient without compromising security. Moreover, quantum encryption is integrated for its strong security features but is not itself a metaheuristic optimisation technique. Metaheuristic techniques are applied to optimise parameters such as key generation and chaotic map configurations, which improve the overall encryption process. The evaluation of our proposed framework through comprehensive statistical analyses revealed impressive results: an entropy value of 7.9998, a near-zero correlation of 0.0001, and an extensive key space of $2^{947.862}$. These metrics show the algorithm's strong encryption capabilities and resistance to various forms of cyberattacks, including noise, cropping, and brute force. The low correlation and high entropy indicate a high level of confusion and diffusion, ensuring the security and integrity of encrypted data. Additionally, the computational complexity analysis reveals that the proposed encryption scheme can encrypt a plaintext image of size 256×256 or 512×512 in under one second. This performance characteristic makes it well-suited for real-time applications and IoT devices. The implementation of the proposed encryption framework on FPGA may face challenges due to high power consumption and processing delays, requiring further optimization for real-time IoT deployment. However, in the future, FPGA-based implementation will be explored to assess real-time performance, energy efficiency, and hardware resource utilisation.

Data availability

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

Received: 27 September 2024; Accepted: 7 April 2025
Published online: 23 April 2025

References

1. Nižetić, S. et al. Internet of things (iot): Opportunities, issues and challenges towards a smart and sustainable future. *J. Clean. Prod.* **274**, 122877 (2020).
2. Chaudhary, A. Internet of things (iot): Research challenges and future applications. *Int. J. Emerg. Trends Sci. Technol.* (2022).
3. Shafique, A. & Ahmed, F. Image encryption using dynamic s-box substitution in the wavelet domain. *Wirel. Pers. Commun.* **115**, 2243–2268 (2020).
4. Tahaei, H., Afifi, F., Asemi, A., Zaki, F. & Anuar, N. B. The rise of traffic classification in iot networks: A survey. *J. Netw. Comput. Appl.* **154**, 102538 (2020).
5. Sheng, Z., Mahapatra, C., Zhu, C. & Leung, V. C. Recent advances in industrial wireless sensor networks toward efficient management in iot. *IEEE Access* **3**, 622–637 (2015).
6. Jamshed, M. A., Ali, K., Abbasi, Q. H., Imran, M. A. & Ur-Rehman, M. Challenges, applications, and future of wireless sensors in internet of things: A review. *IEEE Sens. J.* **22**, 5482–5494 (2022).
7. Bader, J. & Michala, A. L. Searchable encryption with access control in industrial internet of things (iiot). *Wirel. Commun. Mob. Comput.* **2021**, 5555362 (2021).
8. Naru, E. R., Saini, H. & Sharma, M. A recent review on lightweight cryptography in iot. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 887–890 (IEEE, 2017).
9. Gupta, M., Singh, V. P., Gupta, K. K. & Shukla, P. K. An efficient image encryption technique based on two-level security for internet of things. *Multimedia Tools Appl.* **82**, 5091–5111 (2023).
10. Rijmen, V. & Daemen, J. Advanced encryption standard. *Proc. Federal Inf. Process. Stand. Publ. Natl. Inst. Stand. Technol.* **19**, 22 (2001).
11. Standard, D. E. et al. Data encryption standard. *Federal Inf. Process. Standards Publ.* **112**, 3 (1999).
12. Shafique, A., Mehmood, A. & Elhadef, M. Survey of security protocols and vulnerabilities in unmanned aerial vehicles. *IEEE Access* **9**, 46927–46948 (2021).
13. Singh, P., Acharya, B. & Chaurasiya, R. K. A comparative survey on lightweight block ciphers for resource constrained applications. *Int. J. High Perform. Syst. Archit.* **8**, 250–270 (2019).
14. Kumar, R., Kumar, P., Aloqaily, M. & Aljuhani, A. Deep-learning-based blockchain for secure zero touch networks. *IEEE Commun. Mag.* **61**, 96–102 (2022).
15. Shafique, A. A noise-tolerant cryptosystem based on the decomposition of bit-planes and the analysis of chaotic gauss iterated map. *Neural Comput. Appl.* **34**, 16805–16828 (2022).
16. Shafique, A. & Shahid, J. Novel image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps. *Eur. Phys. J. Plus* **133**, 331 (2018).
17. Kizza, J. M. Internet of things (iot): Growth, challenges, and security. In *Guide to Computer Network Security*, 557–573 (Springer, 2024).
18. Singh, S., Sharma, P. K., Moon, S. Y. & Park, J. H. Advanced lightweight encryption algorithms for iot devices: Survey, challenges and solutions. *J. Ambient Intell. Hum. Comput.* 1–18 (2024).
19. Shafique, A. A new algorithm for the construction of substitution box by using chaotic map. *Eur. Phys. J. Plus* **135**, 194 (2020).

20. Madakam, S. & Date, H. Security mechanisms for connectivity of smart devices in the internet of things. In: *Connectivity Frameworks for Smart Devices: The Internet of Things from a Distributed Computing Perspective* 23–41 (2016).
21. Shafique, A., Mehmood, A., Alawida, M., Elhadef, M. & Rehman, M. U. A fusion of machine learning and cryptography for fast data encryption through the encoding of high and moderate plaintext information blocks. *Multimedia Tools Appl.* 1–27 (2024).
22. Alexan, W., Chen, Y.-L., Por, L. Y. & Gabr, M. Hyperchaotic maps and the single neuron model: A novel framework for chaos-based image encryption. *Symmetry* **15**, 1081 (2023).
23. 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).
24. Kumar, S. et al. Chaos based image encryption security in cloud computing. *J. Discrete Math. Sci. Cryptogr.* **25**, 1041–1051 (2022).
25. Zaid, B. et al. Toward secure and resilient networks: A zero-trust security framework with quantum fingerprinting for devices accessing network. *Mathematics* **11**, 2653 (2023).
26. Shafique, A., Mehmood, A. & Elhadef, M. Detecting signal spoofing attack in UAVs using machine learning models. *IEEE Access* **9**, 93803–93815 (2021).
27. Hedayati, R. & Mostafavi, S. A lightweight image encryption algorithm for secure communications in multimedia internet of things. *Wireless Pers. Commun.* **123**, 1121–1143 (2022).
28. İnce, C., İnce, K. & Hanbay, D. Novel image pixel scrambling technique for efficient color image encryption in resource-constrained iot devices. *Multimedia Tools Appl.* 1–29 (2024).
29. Biswas, A., Majumdar, A., Nath, S., Dutta, A. & Baishnab, K. L. Lrbc: A lightweight block cipher design for resource constrained iot devices. *J. Ambient Intell. Hum. Comput.* 1–15 (2023).
30. Mehmood, A. et al. A time-efficient and noise-resistant cryptosystem based on discrete wavelet transform and chaos theory: An application in image encryption. *J. Inf. Secur. Appl.* **78**, 103590 (2023).
31. Guo, L., Du, H. & Huang, D. A quantum image encryption algorithm based on the feistel structure. *Quantum Inf. Process.* **21**, 1–18 (2022).
32. 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).
33. Liu, H., Zhao, B. & Huang, L. A novel quantum image encryption algorithm based on crossover operation and mutation operation. *Multimedia Tools Appl.* **78**, 20465–20483 (2019).
34. Diro, A. et al. Lightweight authenticated-encryption scheme for internet of things based on publish-subscribe communication. *IEEE Access* **8**, 60539–60551 (2020).
35. Kanwal, S. et al. Securing blockchain-enabled smart health care image encryption framework using tinkerbell map. *Alex. Eng. J.* **107**, 711–729 (2024).
36. Inam, S., Kanwal, S., Anwar, A., Mirza, N. F. & Alfraihi, H. Security of end-to-end medical images encryption system using trained deep learning encryption and decryption network. *Egypt. Inf. J.* **28**, 100541 (2024).
37. Inam, S., Kanwal, S., Firdous, R. & Hajjej, F. Blockchain based medical image encryption using Arnold's cat map in a cloud environment. *Sci. Rep.* **14**, 5678 (2024).
38. Jain, K. *et al.* A lightweight multi-chaos-based image encryption scheme for iot networks. *IEEE Access* (2024).
39. Trujillo-Toledo, D. et al. Real-time rgb image encryption for iot applications using enhanced sequences from chaotic maps. *Chaos, Solitons Fractals* **153**, 111506 (2021).
40. Gabr, M. et al. R 3-rescale, rotate, and randomize: A novel image cryptosystem utilizing chaotic and hyper-chaotic systems. *IEEE Access* **11**, 119284–119312 (2023).
41. Shafique, A., Mehmood, A., Elhadef, M. & Khan, K. H. A lightweight noise-tolerant encryption scheme for secure communication: An unmanned aerial vehicle application. *PLoS ONE* **17**, e0273661 (2022).
42. Alexan, W. *et al.* Anteater: When Arnold's cat meets Langton's ant to encrypt images. *IEEE Access* (2023).
43. Alexan, W., El-Damak, D. & Gabr, M. Image encryption based on fourier-DNA coding for hyperchaotic chen system, chen-based binary quantization s-box, and variable-base modulo operation. *IEEE Access* (2024).
44. El-Damak, D. *et al.* Fibonacci q-matrix, hyperchaos, and galois field (2 8) for augmented medical image encryption. *IEEE Access* (2024).
45. Youssef, M. *et al.* Enhancing satellite image security through multiple image encryption via hyperchaos, SVD, RC5, and dynamic s-box generation. *IEEE Access* (2024).
46. Clemente-Lopez, D., de Jesus Rangel-Magdaleno, J. & Muñoz-Pacheco, J. M. A lightweight chaos-based encryption scheme for iot healthcare systems. *Internet of Things* **25**, 101032 (2024).
47. Mondal, B. & Singh, J. P. A lightweight image encryption scheme based on chaos and diffusion circuit. *Multimedia Tools Appl.* **81**, 34547–34571 (2022).
48. Shafique, A., Mehmood, A., Alawida, M., Khan, A. N. & Shuja, J. Lightweight image encryption scheme for iot environment and machine learning-driven robust s-box selection. *Telecommun. Syst.* **88**, 1–23 (2025).
49. Yang, Q., Zhu, D. & Yang, L. A new 7d hyperchaotic system with five positive lyapunov exponents coined. *Int. J. Bifurc. Chaos* **28**, 1850057 (2018).
50. Li, Q. & Chen, L. An image encryption algorithm based on 6-dimensional hyper chaotic system and DNA encoding. *Multimedia Tools Appl.* **83**, 5351–5368 (2024).
51. Zhou, Y., Bao, L. & Chen, C. P. A new 1d chaotic system for image encryption. *Signal Process.* **97**, 172–182 (2014).
52. Haspolat, E. & Yıldız, B. Fractional order of a new 7d hyperchaotic lorenz-like system. *Konuralp J. Math.* **9**, 76–89 (2021).
53. Shukla, P. D. Complex wavelet transforms and their applications. In: *A Dissertation Submitted of Signal Processing Division, Department of Electronic and Electrical Engineering University of Strathclyde Scotland United Kingdom* (2003).
54. Shafique, A., Mehmood, A., Alawida, M., Khan, A. N. & Khan, A. A novel machine learning technique for selecting suitable image encryption algorithms for iot applications. *Wirel. Commun. Mob. Comput.* **2022**, 5108331 (2022).
55. Shafique, A., Hazzazi, M. M., Alharbi, A. R. & Hussain, I. Integration of spatial and frequency domain encryption for digital images. *IEEE Access* **9**, 149943–149954 (2021).
56. Kamal, S. T., Hosny, K. M., Elgindy, T. M., Darwish, M. M. & Fouda, M. M. A new image encryption algorithm for grey and color medical images. *Ieee Access* **9**, 37855–37865 (2021).
57. Shafique, A. & Ahmed, J. Dynamic substitution based encryption algorithm for highly correlated data. *Multidimension. Syst. Signal Process.* **32**, 91–114 (2021).
58. Hao, W., Zhang, T., Chen, X. & Zhou, X. A hybrid neqr image encryption cryptosystem using two-dimensional quantum walks and quantum coding. *Signal Process.* **205**, 108890 (2023).
59. Huang, X., Dong, Y., Ye, G. & Shi, Y. Meaningful image encryption algorithm based on compressive sensing and integer wavelet transform. *Front. Comput. Sci.* **17**, 173804 (2023).
60. Yang, Y., Cheng, M., Ding, Y. & Zhang, W. A visually meaningful image encryption scheme based on lossless compression spiht coding. *IEEE Trans. Serv. Comput.* **16**, 2387–2401 (2023).
61. Mehmood, A., Shafique, A., Kumar, N. & Bhutta, M. N. Data security in the industrial internet of things (iiot) through a triple-image encryption framework leveraging 3-d neat, 1dcj, and 4dhcfo techniques. *Comput. Electr. Eng.* **118**, 109354 (2024).
62. Abed, Q. K. & Al-Jawher, W. A. M. A secure and efficient optimized image encryption using block compressive sensing and logistic map method. *J. Cyber Secur. Mobil.* 983–1006 (2024).

63. Rehman, M. U. & Shafqiue, A. Robust encryption framework for iot devices based on bit-plane extraction, chaotic sine models, and quantum operations. *Internet of Things* 101241 (2024).
64. Kocak, O., Erkan, U., Toktas, A. & Gao, S. Pso-based image encryption scheme using modular integrated logistic exponential map. *Expert Syst. Appl.* **237**, 121452 (2024).
65. Alvarez, G. & Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurc. Chaos* **16**, 2129–2151 (2006).
66. Dhar, S., Khare, A., Dwivedi, A. D. & Singh, R. Securing iot devices: A novel approach using blockchain and quantum cryptography. *Internet of Things* **25**, 101019 (2024).
67. Kuldeep, G. & Zhang, Q. Design prototype and security analysis of a lightweight joint compression and encryption scheme for resource-constrained iot devices. *IEEE Internet Things J.* **9**, 165–181 (2021).
68. Iftikhar, Z. et al. Privacy preservation in resource-constrained iot devices using blockchain-a survey. *Electronics* **10**, 1732 (2021).
69. Khrennikov, A. Roots of quantum computing supremacy: Superposition, entanglement, or complementarity?. *Eur. Phys. J. Spec. Top.* **230**, 1053–1057 (2021).
70. Khang, A., Rath, K. C., Panda, N. & Kumar, A. Quantum mechanics primer: Fundamentals and quantum computing. In *Applications and Principles of Quantum Computing*, 1–24 (IGI Global, 2024).
71. Shafique, A. et al. A hybrid encryption framework leveraging quantum and classical cryptography for secure transmission of medical images in iot-based telemedicine networks. *Sci. Rep.* **14**, 31054 (2024).

Acknowledgements

The authors extend their appreciation to the Deanship of Research and Graduate Studies at University of Tabuk for funding this work through Research No. S-1444-0107

Author contributions

A.A formal analysis and supervision, A.R.A conceptualization and editing, A.A. methodology, software, and writing, M.K.A. validation, S.A. review, A.S visualization and co-supervision. All authors reviewed the manuscript.

Declarations

Competing interests

No, I declare that the authors have no competing interests as defined by Nature Research, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Additional information

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