Ingénierie des Systèmes d'Information
Vol. 30, No. 10, October, 2025, pp. 2691-2705
Journal homepage: http://iieta.org/journals/isi

Quantum Image Encryption Using 4D Hamiltonian System and Bit-Plane Encoding

Ahmed M. Ajaj[1], Mayada Taki Wazi[2], Farah J. Al-Zahid[3], Nadia M. G. Al-Saidi[4*]

[1] Department of Islamic Banking and Finance, College of Islamic Sciences, Aliraqia University, Baghdad 10053, Iraq
[2] College of Electromechanical Engineering, University of Technology, Baghdad 10066, Iraq
[3] Department of Mathematics, College of Science for Women, University of Baghdad, Baghdad 10071, Iraq
[4] College of Applied Sciences, University of Technology, Baghdad 10066, Iraq

Corresponding Author Email: nadia.m.ghanim@uotechnology.edu.iq

https://doi.org/10.18280/isi.301015

Received: 27 July 2025
Revised: 11 September 2025
Accepted: 20 September 2025
Available online: 31 October 2025

Keywords:
cryptosystem, image encryption, quantum image representation, quantum cryptography, chaotic system, 4D quantum hyperchaotic system, QIRBP, quantum circuit depth

ABSTRACT

This paper presents a new quantum image encryption (QIE) algorithm that integrates a newly developed 4D Quantum Logistic-Jerk Hyperchaotic System (4D-QLJHS) with a quantum image representation (QIR) model, enabling secure and efficient image transmission. By reducing qubit usage and circuit depth, the proposed QIE framework significantly improves quantum resource efficiency. Additionally, incorporating the enhanced 4D-LJHS map strengthened security and resistance against both quantum and classical attacks, surpassing the robustness and scalability of current quantum image encryption techniques. The new 4D-QLJHS is created by combining a logistic map and a jerk system, and then converted to a quantum hyperchaotic system using a Hamiltonian-based method. These systems help create a Quantum Pseudo-Random Number Generator (QPRNG) that generates random bit sequences used to alter and rearrange data at the quantum bit-plane level during the encryption process. The quantum encryption method uses the Quantum Image Representation based on Bit Planes (QIRBP) model, which enables modifying individual pixels and color channels via CNOT and SWAP gates. We evaluate various statistical tests to confirm the security and efficiency of our system. The experiments demonstrate that the system is highly secure, featuring adequate randomness, robust protection against specific attacks, and uniform distribution in the encrypted image data. Experimental simulations of some images indicate that the system provides a satisfactory level of security for image encryption, given the computational costs. This makes it suitable for real-time image communication where security is a priority.

1. INTRODUCTION

As the digital age continues to advance, images are a crucial means of conveying information [1]. They have become increasingly prevalent across domains such as financial transactions, military reconnaissance, and medical imaging. However, several obstacles and issues need to be addressed using conventional image encryption technology [2]. First, the threat of quantum computation, along with the powerful computational capabilities of quantum computers, presents serious challenges to the security of traditional encryption schemes. First, the strong computational capabilities of quantum computers pose a serious threat to the security of encryption algorithms based on conventional mathematical puzzles (such as discrete logarithms and integer factorization), and the security of classical encryption algorithms faces major challenges when confronted with quantum computation [3]. The approach developed by Grover significantly reduces the security of current encryption methods; for instance, it can halve the work required for key search in symmetric encryption algorithms [4]. These traditional encryption techniques can also be successfully cracked in polynomial time using Shor's method [5]. The second type of encryption makes it difficult for conventional encryption methods, such as stream ciphers, to efficiently interpret image data, since it often exhibits high redundancy and strong correlation. Furthermore, this redundancy can easily lead to the loss of local information during the encryption process, jeopardising the overall security of the encryption. The introduction of deep learning-based ciphertext analysis tools and quantum-computing-accelerated brute-force cracking has put the security of traditional encryption algorithms to the ultimate test [6]. Information security is increasingly threatened by deep learning models that may be trained on vast amounts of data to progressively discover and exploit flaws in encryption techniques.

Three categories of QIR can be distinguished: chaos-based, transform-domain-based, and spatial domain-based. Researchers have used the spatial and transform domains of quantum computers to create image encryption methods. In 2012, Zhou et al. [7] proposed a quantum picture encryption method based on geometric transformations of quantum images. QIR, based on limited geometric and color modifications, was introduced by Song et al. [8]. A QIR technique based on the Arnold transform was presented by Zhou et al. [9]. A QIR technique based on image correlation decomposition was proposed by Hua et al. [10].

With the rapid advancement of quantum computing hardware, experimental and theoretical investigations into quantum random walks will advance, and these studies are projected to become increasingly important in the fields of quantum computing, quantum information, and quantum simulation in the future [11]. Quantum random walks have been applied to image encryption to induce pixel disruption or generate dynamic keys. To accomplish color image encryption, Zhao et al. [12] used the tessellation transform in conjunction with quantum random walks. However, the periodicity of chaotic systems limits the security, and key generation depends on classical chaotic mappings. Panda and Benjamin [13] developed encryption techniques based on quantum random walks and DNA coding. They, however, did not fully utilize quantum parallelism. In this work, we aim to address the issue of QIE techniques that underutilize quantum features to encrypt large images and rely on classical keys, thereby reducing security.

Most QIE models demand excessive qubits (often > 20) and complex gate sequences and quantum operations, exceeding the capabilities of near-term (Noisy Intermediate-Scale Quantum, NISQ) devices [14]. For instance, some models need more than 30 qubits to represent a basic image, rendering them infeasible for current hardware. Many encryption algorithms are based on chaotic systems with inherent periodicity, thereby weakening resistance to differential attacks and failing to leverage quantum properties, such as entanglement, to counter quantum threats like Grover’s algorithm [4, 5]. Moreover, a few models address deep learning-based cryptanalysis, leaving encrypted images vulnerable to pattern recognition exploits [6]. Currently existing systems lack scalability for large pictures or real-time transmission. Additionally, current quantum parallelism leads to inefficient circuits, where processing moderate-sized images (e.g., 128 × 128 pixels) results in either restricted execution times or high error rates due to qubit coherence [13, 15].

Here, we describe a new QIE system in this research that combines a QIR model based on bit planes (QIRBP) [14] with a 4D-LJHS. To increase the system's complexity, the 4D-LJHS was enhanced by introducing nonlinear terms into the classical logistic and jerk dynamics. We then converted this system to the quantum domain using a Hamiltonian-based quantization approach, resulting in a 4D quantum logic junction (QLJHS). The quantum evolution governed by this Hamiltonian system was used to build a quantum pseudo-random number generator (QPRNG), which drives substitution and permutation operations at the bit-plane level. By integrating the 4D-QLJHS with the QIRBP model, we developed a QIE framework that ensures high randomness, strong resistance to differential attacks, and efficient circuit performance. The motivation behind this approach is to provide a secure, scalable, and real-time quantum encryption algorithm suitable for practical image transmission over quantum communication systems.

In this paper, the following contribution is made:

- A new 4D Logistic–Jerk Hyperchaotic System (4D-LJHS) was designed by integrating a logistic map with Jerk chaotic systems. Nonlinear steps added to some system parameters increased complexity and made encryption algorithms more secure.
- A new 4D Quantum Hyperchaotic Logistic–Jerk System was designed by converting a classical 4D-LJHS to a quantum state via a quantization system.
- Introducing a new diffusion method that combines chaotic sequences with a diffusion process driven by quantum pseudorandom numbers.
- Designing a new cryptosystem with real-time applications supporting high security and low complexity to be used for a rapid image encryption algorithm by the proposed 4D-QHLJS for fast and secure image transmission.
- Analyzing the results in terms of performance and security by utilizing some evaluation parameters such as the NIST 800-22 randomness test [16], histogram analysis, entropy calculations, and correlation-based approach. The retrieved results emphasize that the newly presented system in this article is highly secured against differential and statistical attacks.

The remainder of the paper is structured as follows: Section 2 provides a detailed overview of chaotic systems and their applications in cryptography. This section introduces the mathematical background of the Logistic and Jerk chaotic systems and their quantum systems, which form the foundation of the proposed 4D-QLJHS. In the same section, we present the system's mathematical formulation and discuss its dynamic behavior, including attractors and Lyapunov exponents. The complexity and unpredictability of the system are demonstrated through experimental results. Section 3 introduces the image encryption algorithm, outlining the diffusion and confusion processes in detail. This section includes the pseudocode for the encryption and decryption algorithms, providing a step-by-step description of how the proposed 4D-QLJHS is used to secure image data. Section 4 presents the security analysis of the proposed cryptosystem. We conduct several experiments to evaluate the system's performance in terms of key sensitivity, randomness, entropy, and resistance to differential attacks. The results demonstrate that the proposed system outperforms existing chaos-based encryption schemes in both security and efficiency. Finally, the paper is concluded in Section 5 by summarizing the key findings and discussing potential future work.

2. THEORETICAL CONCEPTS

This section introduces a new 4D chaotic system. After that, it converts this system from a classical state to a quantum state via a quantization process.

2.1 Generating a new hyperchaotic system

A new 4D chaotic system is introduced in this study. We derived this system by extending the logistic map [17] and from the classical three-dimensional Jerk chaotic system [18]. By incorporating changes in parameters and nonlinear terms, while preserving the original dynamics, we formulated a novel 4-dimensional hyperchaotic mathematical model. This process involved introducing new variables represented as 'w', adjusting nonlinear terms, eliminating unfavourable components, and ensuring consistency with forward changes. Its chaotic behavior and complexity evaluation are investigated by experimental evidence. The following equations define the 4D chaotic system:

$$\begin{aligned} \frac{dx}{dt} &= \beta y \\ \frac{dy}{dt} &= \beta z \\ \frac{dz}{dt} &= -ay - bz - w + g\exp^{(dx-ex^2)} \\ \frac{dw}{dt} &= \alpha x + \beta z(1-z) \end{aligned} \qquad (1)$$

2.2 The dynamic behaviour of the proposed chaotic system

An "attraction" in a dynamical system is a collection of points or a sequence of values in state space that indicate the output's path of travel. We can maintain the ergodicity of the dynamical system and ensure the hyperchaotic structure. Figure 1 shows the attractor diagram with different values of the initial states $(x, y, z, w) = (1{,}1{,}1{,}1)$, and the parameters for Figure 1(a) are $(a = 50.60,\ b = 0.0001,\ d = 0.01,\ e = 0.005,\ \alpha = 1.4,\ \beta = 1.5)$, for Figure 1(b) $(a = 90.60,\ b = 1.1,\ d = 3.10,\ e = 10.850,,\ g = 50.1,\ \alpha = 1.4,\ \beta = 1.5)$, for Figure 1(c) $(a = 60.60,\ b = 0.001,\ d = 0.01,\ e = 0.850,\ g = 0.1,\ \alpha = 10.4,\ \beta = 10.5)$, for Figure 1(d) are $(a = 60.60,\ b = 0.0001,\ d = 0.1,\ e = 1.85,\ g = 5.1,\ \alpha = 10.4,\ \beta = 10.5)$, for Figure 1(e) are $(a = 90.60,\ b = 0.1,\ d = 0.1,\ e = 1.85,\ g = 5.1,\ \alpha = 10.4,\ \beta = 10.5)$, and Figure 1(f) $(a = 90.60,\ b = 1.1,\ d = 3.1,\ e = 1.85,\ g = 5.1,\ \alpha = 10.4,\ \beta = 10.5)$, and the time step starts at time 0 and simulates up to time 500. Figure 2 shows the sensitivity of the system (1), with the fixed parameters $(a = 90.60,\ d = 3.1,\ e = 1.85,\ g = 5.1,\ \alpha = 10.4,\ \beta = 10.5)$ and we change b value 0.1 in each state where (a, b, c) the value of $b = (0.1, 0.3, 0.5)$, respectively.

2.2.1 Dissipative of system (1)

The divergence of the system (1) can be calculated via the following equation:

$$\nabla f = \frac{\partial f_{x_1}}{\partial x} + \frac{\partial f_{x_2}}{\partial y} + \frac{\partial f_{x_3}}{\partial z} + \frac{\partial f_{x_4}}{\partial x} = -b \qquad (2)$$

where, $f_x = \dot{x}, f_y = \dot{y}, f_z = \dot{z}$. Since the divergence of the system (1) is negative for all positive values of c, where $b > 0$, then the proposed system (1) has dissipative behaviour.

2.2.2 Lyapunov exponents

The Lyapunov Exponents $(\mathcal{LE})$ is defined as a measure of how quickly nearby trajectories either diverge from or converge toward each other. It can be defined as [15]:

$$\lambda_{Le} \cong \frac{1}{t}\ln\frac{\| \delta x(t) \|}{\| \delta x(0) \|} \qquad (3)$$

where, $\frac{\|x(t)\|}{\|x(0)\|}$ refers to the distance between two different trajectories. Figure 3 shows the $(\mathcal{LE})$ of the system (1), where the initial states $(x, y, z, w) = (0.1{,}0.1{,}0.1{,}0.1)$, and the parameters are $(a = 50.60,\ b = 0.0001,\ d = 0.01,\ e = 0.005,\ f = 0.1,\ g = 0.1,\ \alpha = 1.4,\ \beta = 1.5)$ respectively.

2.2.3 Permutation entropy

Given a set window length and relying on the distribution of these permutation patterns, permutation entropy (PE) is the entropy of a random variable that samples the occurrences of permutation patterns from a time series. Figures 4(a) and (b) show the PE of the system(1), where the parameters $(a = 90.5,\ b = 60.5,\ d = 10.8,\ e = 0.85,\ \alpha = 1.4,\ \beta = 1.5)$ and $(a = 1.5,\ b = 1.5,\ d = 10.8,\ e = 0.85,\ \alpha = 0.4,\ \beta = 0.5)$, respectively.

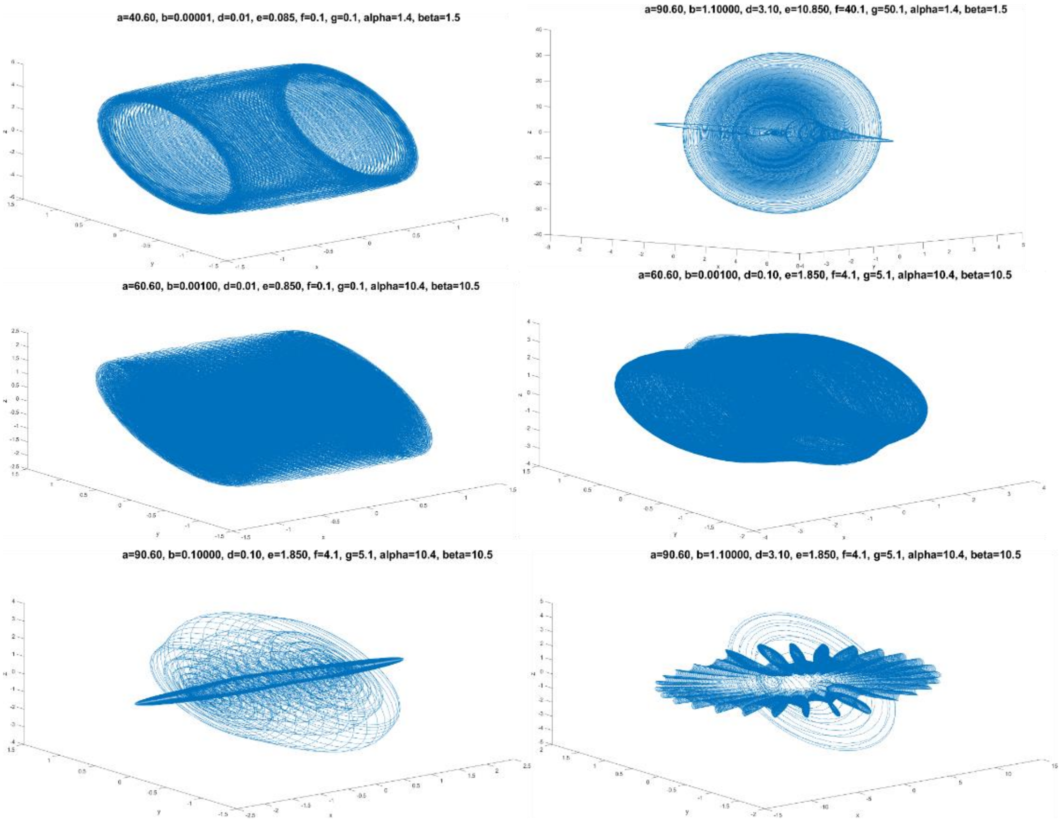
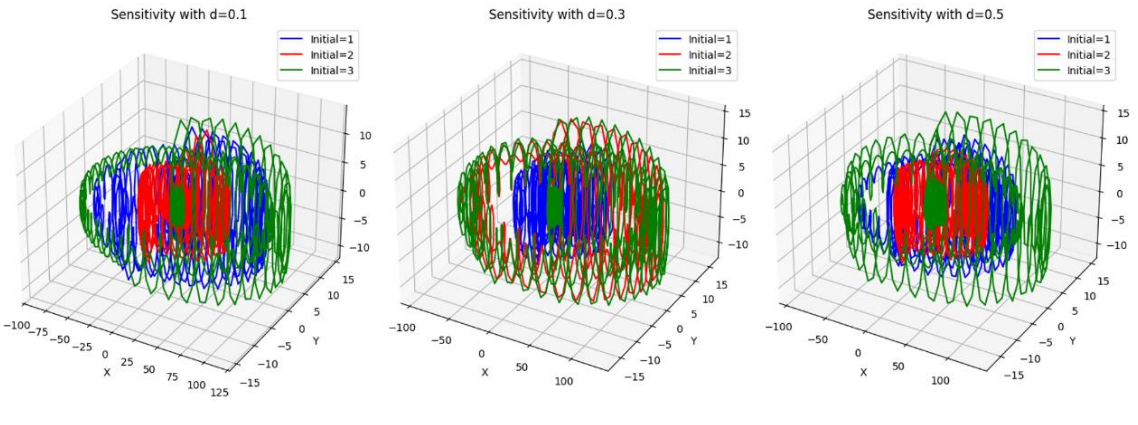


Figure 1. Attractors of the system (1)



(a)

(b)

(c)

Figure 2. The sensitivity of the system (1)

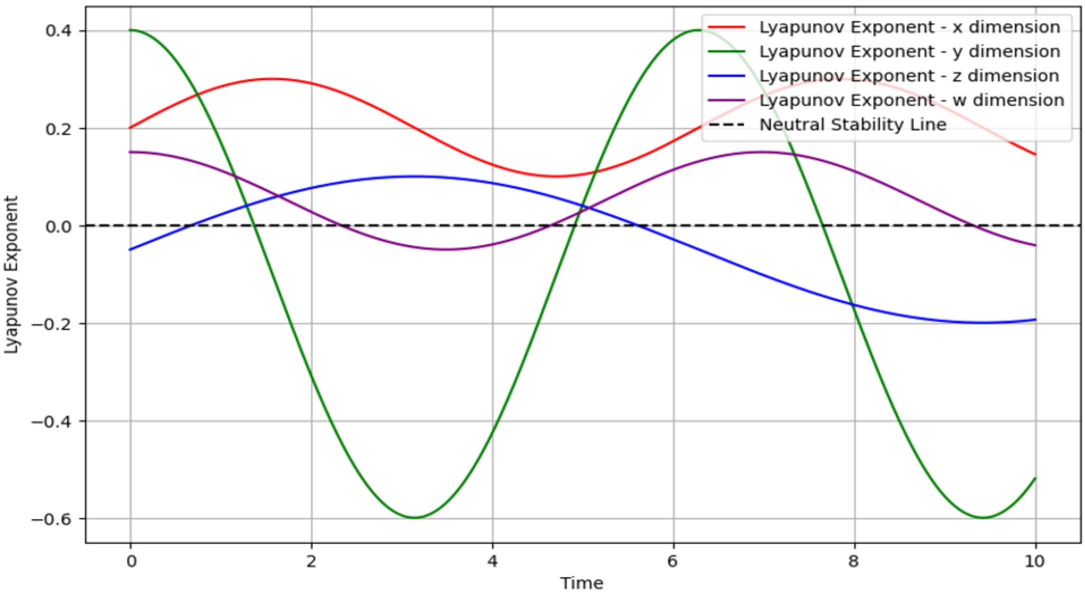
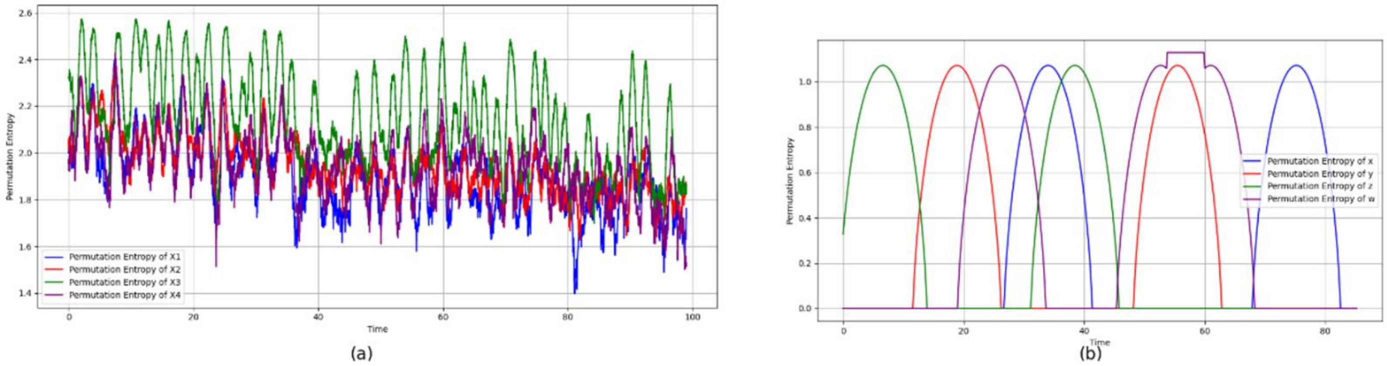


Figure 3. Lyapunov Exponents of the system (1)



(a)

(b)

Figure 4. Permutation entropy for all variables

Figure 4 presents the permutation entropy (PE) of the system (1) under two different parameter sets. The graph illustrates how the system's chaotic behavior changes with variations in the parameters. Specifically, PE measures the complexity and unpredictability of the system, with higher values indicating more chaotic dynamics. This analysis highlights the system's potential for secure cryptographic applications.

3. QUANTIZATION SYSTEM

Based on the reference [19] in this method, we will convert (1) from a classical state to a quantum state via a quantization system, as our system is dissipative. First step, we will find the Hamiltonian formulation. We extend the state space by introducing canonically conjugate momenta (p_x, p_y, p_z, p_w) corresponding to each variable. The dynamics can be encoded by a Hamiltonian function $H_{\mathrm{cl}}(x, y, z, w, p_x, p_y, p_z, p_w)$

defined as:

$$\begin{aligned} H_{\mathrm{cl}} = \beta y\, p_x + \beta z\, p_y \\ + \left(-ay - bz - w + g\, e^{dx-ex^2}\right) p_z \\ + (\alpha x + \beta z(1-z)) p_w \end{aligned} \tag{4}$$

The system evolves according to Hamilton's canonical equations:

$$\frac{dq_i}{dt} = \frac{\partial H_{\mathrm{cl}}}{\partial p_i}, \frac{dp_i}{dt} = -\frac{\partial H_{\mathrm{cl}}}{\partial q_i}, i \in \{x, y, z, w\}. \tag{5}$$

From this Hamiltonian, one recovers the original system as the configuration-space evolution:

$$\begin{aligned} \frac{dx}{dt} &= \frac{\partial H_{\mathrm{cl}}}{\partial p_x} = \beta y, \\ \frac{dy}{dt} &= \frac{\partial H_{\mathrm{cl}}}{\partial p_y} = \beta z, \\ \frac{dz}{dt} &= \frac{\partial H_{\mathrm{cl}}}{\partial p_z} = -ay - bz - w + g\, e^{dx-ex^2}, \\ \frac{dw}{dt} &= \frac{\partial H_{\mathrm{cl}}}{\partial p_w} = \alpha x + \beta z(1-z), \end{aligned} \tag{6}$$

Thus, confirming consistency with the original flow.

In the second step, we promote the phase-space variables to operators working on a Hilbert space in order to quantise the classical system $\mathcal{H}$. Define:

$$x \mapsto \hat{x}, y \mapsto \hat{y}, z \mapsto \hat{z}, w \mapsto \hat{w},$$

$$p_x \mapsto \hat{p}_x = -i\hbar \frac{\partial}{\partial x}, \text{etc.}$$

Impose canonical commutation relations:

$$\begin{aligned} [\hat{q}_i, \hat{p}_j] = i\hbar\, \delta_{ij}, \text{with } \hat{q}_i, \hat{p}_j \\ \in \{\hat{x}, \hat{y}, \hat{z}, \hat{w}, \hat{p}_x, \hat{p}_y, \hat{p}_z, \hat{p}_w\}. \end{aligned} \tag{7}$$

To ensure Hermiticity, we use Weyl (symmetric) ordering. The quantum Hamiltonian operator $\hat{H}$ is thus constructed as:

$$\hat{H} = \frac{1}{2} \sum_{i \in \{x,y,z,w\}} \left(f_i(\hat{x}, \hat{y}, \hat{z}, \hat{w})\, \hat{p}_i + \hat{p}_i\, f_i(\hat{x}, \hat{y}, \hat{z}, \hat{w})\right), \tag{8}$$

with:

$$\begin{aligned} f_x &= \beta \hat{y}, \\ f_y &= \beta \hat{z}, \\ f_z &= -a\hat{y} - b\hat{z} - \hat{w} + g\, e^{d\hat{x}-e\hat{x}^2}, \\ f_w &= \alpha \hat{x} + \beta \hat{z}(1-\hat{z}). \end{aligned} \tag{9}$$

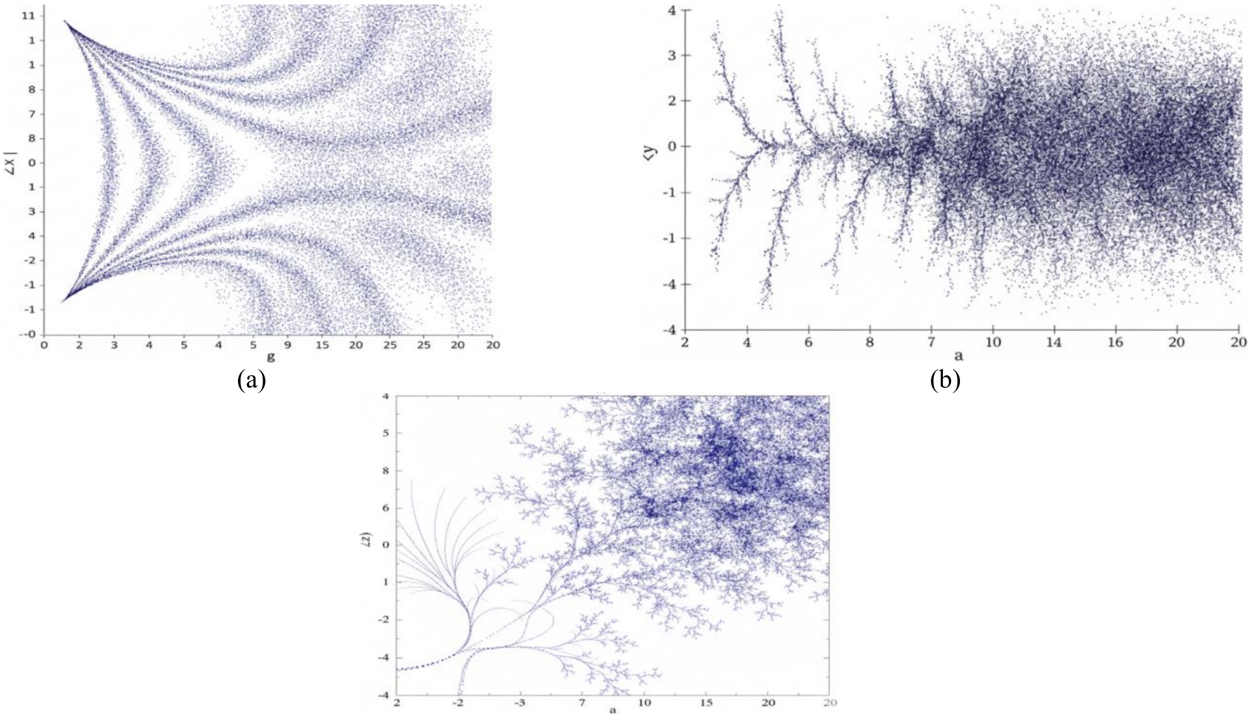
This leads to the explicit expression:

$$\begin{aligned} \hat{H} = \frac{1}{2}\beta(\hat{y}\hat{p}_x + \hat{p}_x\hat{y}) + \frac{1}{2}\beta(\hat{z}\hat{p}_y + \hat{p}_y\hat{z}) \\ + \frac{1}{2}\begin{bmatrix} \left(-a\hat{y} - b\hat{z} - \hat{w} + g\, e^{d\hat{x}-e\hat{x}^2}\right)\hat{p}_z + \\ \hat{p}_z\left(-a\hat{y} - b\hat{z} - \hat{w} + g\, e^{d\hat{x}-e\hat{x}^2}\right) \end{bmatrix} \\ + \frac{1}{2}\begin{bmatrix} \left(\alpha\hat{x} + \beta\hat{z}(1-\hat{z})\right)\hat{p}_w + \\ \hat{p}_w\left(\alpha\hat{x} + \beta\hat{z}(1-\hat{z})\right) \end{bmatrix} \end{aligned} \tag{10}$$

This Hermitian operator $\hat{H}$ generates dynamics through the Heisenberg equation of motion:

$$\frac{d\hat{O}}{dt} = \frac{i}{\hbar}[\hat{H}, \hat{O}], \forall \hat{O} \in \{\hat{x}, \hat{y}, \hat{z}, \hat{w}, \hat{p}_x, \dots\} \tag{11}$$

Figure 5 shows the bifurcation of (10) for several dimensions (x, y, z) vs. perimeter a.



(a)

(b)

(c)

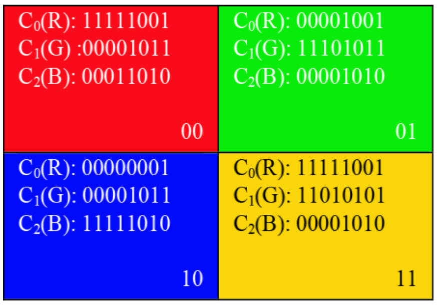
Figure 5. Bifurcation of (10), (a) x vs. a (b) y vs. a (c) z vs. a

4. QUANTUM IMAGE REPRESENTATION

The idea of QIR, which transforms classical image data into quantum states for effective processing, manipulation, and storage on quantum computers, is presented in this subsection. QIR models aim to preserve image characteristics, such as pixel positions, color information, and bit-plane decomposition, within a compact quantum structure. One such advanced model is the QIRBP, which provides a fine-grained decomposition of image data across color channels and bit planes, allowing both high-resolution encoding and efficient quantum operations. The QIRBP model is defined as:

$$\left|\mathcal{J}_{QIRBP}\right\rangle = \frac{1}{\sqrt{M}}\sum_{\lambda=0}^{c-1}\sum_{L=0}^{b-1}\sum_{YX=0}^{N-1} |C_{\lambda,L,YX}\rangle \otimes |\lambda\rangle \otimes |L\rangle \otimes |YX\rangle \qquad (12)$$

where, $\lambda \in \{0,1,\ldots,c-1\}$ for color channel (RGB), need 3 qubits for representation of color 2^3, $L \in \{0,1,\ldots,b-1\}$ bit-plane index, $YX \in \{0,1,\ldots,2^{2n}-1\}$: pixel position encoded over $2n$ qubits and $C_{\lambda,L,YX} \in \{0,1\}$ pixel bit value needs a signal qubit. The image size is $2^n \times 2^n = 2^1 \times 2^1 = 2 \times 2$ pixels, as shown in Figure 6, assuming that each colour channel (Red, Green, and Blue) is represented using b=8 bits. If we use the binary representation of the pixel intensities, the QIR for this colour image is as follows:

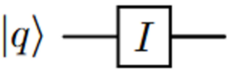
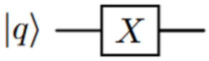
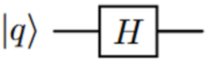
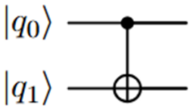
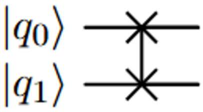


C$_0$(R): 11111001 C$_1$(G) :00001011 C$_2$(B): 00011010 00	C$_0$(R): 00001001 C$_1$(G): 11101011 C$_2$(B): 00001010 01
C$_0$(R): 00000001 C$_1$(G): 00001011 C$_2$(B): 11111010 10	C$_0$(R): 11111001 C$_1$(G): 11010101 C$_2$(B): 00001010 11

Figure 6. A simple 2 × 2 image and its QIRBP state [14]

$$\begin{aligned}|I\rangle = \frac{1}{256}[\square|11111001\rangle \otimes |000\rangle \otimes |11111001\rangle \otimes |00\rangle + |00001011\rangle \otimes |001\rangle \otimes |00001011\rangle \otimes |00\rangle \\ + |00011010\rangle \otimes |010\rangle \otimes |00011010\rangle \otimes |00\rangle + |00001001\rangle \otimes |000\rangle \otimes |00001001\rangle \otimes |01\rangle \\ + |11110101\rangle \otimes |001\rangle \otimes |11110101\rangle \otimes |01\rangle + |00010110\rangle \otimes |010\rangle \otimes |00010110\rangle \otimes |01\rangle + \\ |00000001\rangle \otimes |000\rangle \otimes |00000001\rangle \otimes |10\rangle + |00001011\rangle \otimes |001\rangle \otimes |00001011\rangle \otimes |10\rangle + |11111010\rangle \otimes |010\rangle \\ \otimes |11111010\rangle \otimes |10\rangle + \\ |11111001\rangle \otimes |000\rangle \otimes |11111001\rangle \otimes |11\rangle + |11010101\rangle \otimes |001\rangle \otimes |11010101\rangle \otimes |11\rangle + |00001010\rangle \otimes |010\rangle \\ \otimes |00001010\rangle \otimes |11\rangle].\end{aligned}$$

Table 1. Examples of quantum circuits and their matrix representations

Name	Quantum Circuit	Matrix Formula
Identity I_2		$\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$
Pauli-X		$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$
Hadamard H		$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$
CNOT		$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}$
SWAP		$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 0 & 1 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\end{bmatrix}$

4.1 Quantum gates

Quantum computing uses qubits, physical systems governed by quantum mechanics, to encode and process information. The state of a qubit is represented by a unit vector in a 2-D Hilbert space (H^2), which is represented by the notation $\rangle$. A qubit $|\varphi\rangle$ may be written in general form as:

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha\begin{bmatrix}1\\0\end{bmatrix} + \beta\begin{bmatrix}0\\1\end{bmatrix} = [\alpha \quad \beta]^{\mathrm{T}}$$

where, $|0\rangle$ and $|1\rangle$ represent the basis states in H^2, and $\alpha, \beta \in C$(the complex set), and represent the amplitude of the corresponding computational basis states $|0\rangle$ and $|1\rangle$ that satisfies the normalization condition $|\alpha|^2 + |\beta|^2 = 1$.

The tensor product, denoted by $\otimes$, combines two matrices into a larger block matrix. If Q is an $n \times n$ matrix and Z is an $m \times m$ matrix, then their tensor product $Q \otimes Z$ results in an $nm \times nm$ block matrix is defined as follows:

$$\mathbb{Q} \otimes \mathbb{Z} = \begin{bmatrix}\mathbb{Q}_{0,0}\mathbb{Z} & \cdots & \mathbb{Q}_{0,n-1}\mathbb{Z}\\ \vdots & \vdots & \vdots\\ \mathbb{Q}_{n-1,0}\mathbb{Z} & \cdots & \mathbb{Q}_{n-1,n-1}\mathbb{Z}\end{bmatrix}$$

Small vector spaces can be combined to create a larger vector space through the tensor product. For example, let $|i\rangle$ represent a basis state in a 2^n-dimensional Hilbert space. Each state $|i\rangle$ (for $i = 0,1,2,\dots,2^n - 1$) is formed by the tensor product of n computational basis states:

$$\begin{aligned}|i\rangle = |i_{n-1}\rangle \otimes |i_{n-2}\rangle \otimes \cdots \otimes |i_1\rangle \otimes |i_0\rangle \\ = |i_{n-1}\rangle |i_{n-2}\rangle \dots |i_1\rangle |i_0\rangle \\ = |i_{n-1} i_{n-2} \cdots i_1 i_0\rangle\end{aligned}$$

where, $i = \sum_{j=0}^{n-1} i_j \times 2^j, i_0, i_1, \dots, i_{n-1} \in \{0,1\}$. As a result, the n-qubit's quantum system $|\psi\rangle$ can be defined as a superposition state of 2^nquantum basic states:

$$|\psi\rangle = \sum_{k=0}^{n-1} a_k |k\rangle, k = k_{n-1} k_{n-2} \dots k_1 k_0, k_i \in \{0,1\}$$

and also satisfy $\sum_{k=0}^{n-1} |a_k|^2 = 1$. The components required to build a quantum circuit are quantum gates. One-qubit and two-qubit gates can be used to simulate a complex quantum gate [3,46]. The n-qubit's quantum gate can be represented by a $2^n \times 2^n$unitary matrix. Table 1 presents examples of basic gates and the matrices that accompany them.

4.2 The proposed system for generating pseudo random number sequences

In this section, a QPRNG is introduced based on the time evolution governed by our proposed system in Eq. (10). The evolution of this Hamiltonian is modelled using a gate-based quantum circuit, and the resulting dynamics are exploited to generate high-entropy pseudo-random binary sequences. These sequences are later utilized in the encryption process to enhance randomness and security in QIR. The generation process consists of the following stages: qubit initialization, superposition construction using Hadamard gates, gate-based simulation of a quantum hyperchaotic system, and final projective measurement. Let the total number of qubits be defined as $Q = 4$, where each qubit encodes one of the state variables x, y, z, w in the hyperchaotic system. Steps below show the QPRNG generating:

Step 1: Initialization

All qubits are initially prepared in the computational basis state:

$$|\psi_0\rangle = |0\rangle^{\otimes 4} = |0000\rangle \tag{13}$$

To ensure finding all superpositions equal probability over all input states, apply Hadamard gates H to each qubit. The H is defined as:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{14}$$

$$|\psi_1\rangle = H^{\otimes 4} |\psi_0\rangle = \frac{1}{\sqrt{16}} \sum_{k=0}^{15} |k\rangle \tag{15}$$

This creates a uniform superposition over all 2^4 possible configurations of x, y, z, w.

Step 2: Simulating 4D Hyperchaotic Dynamics via Quantum Gates

The quantum circuit then simulates the dynamics of the Hamiltonian using parameterized rotation for x vector and z vector (RX and RZ), respectively. Controlled rotation gates CRZ and CRY, without relying on operator-based decompositions. Now, based on (10), we have:

Each term is approximately using gate sequences as follows:

- Single-qubit RX and RZ gates simulate local kinetic and phase evolution.
- CRZ gates simulate inter-variable couplings like $\hat{y}\hat{p}_x$, $\hat{z}\hat{p}_y$, $\hat{z}(1-\hat{z})\hat{p}_w$, etc.
- CRX and CRY gates introduce entanglement and nonlinearity into the system.
- Exponential and polynomial terms like $e^{d\hat{x} - e\hat{x}^2}$ are approximated using fixed-angle CRZ gates with control from $\hat{x}$.

This layered gate-based evolution directly implements the system's hyperchaotic interactions.

Step 3: Measurement

After applying T layers of chaotic gate operations, the quantum state is measured:

$$|\psi_T\rangle \xrightarrow{\text{Measure}} |m\rangle, m \in \{0,1\}^4 \tag{16}$$

The measured bitstring $m = m_0 m_1 m_2 m_3$ is the output of a single iteration of the QPRNG. The randomness in m reflects the underlying chaotic quantum evolution.

Step 4: Iteration

The process is repeated until the quantum pseudo-random number sequence QPRNS reaches the desired length N:

$$\text{QPRNS} = \left\{ m^{(1)}, m^{(2)}, \dots, m^{\left(\frac{N}{4}\right)} \right\} \tag{17}$$

Algorithm 1 and Figure 7 show the generation of QPRNG based on 4D-QLJHS.

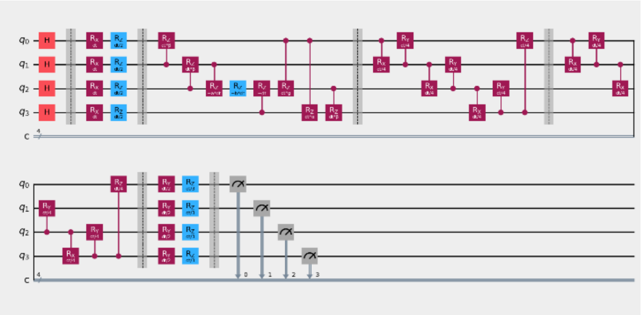


Figure 7. Quantum circuit for QPRNG based on system (10)

This circuit provides a full realization of the QPRNG based on gate-level simulation of the 4D-QLJHS. The randomness of the output is a direct consequence of the nonlinear entanglement and chaotic dynamics encoded in the Hamiltonian evolution.

4.3 The NIST test of the PRNG

The National Institute of Standards and Technology (NIST) offers a comprehensive statistical testing suite designed to evaluate the randomness characteristics of binary sequences. In this section, we employed the NIST SP 800-22 test suite to assess the stochastic properties of the PRNS generated by our proposed QPRNG model based on the 4D-QLJHS.

Algorithm 1. QPRNG Based on 4D-QLJHS

```
Input:
• Evolution depth T (number of layers)
• Number of pseudo-random bits N
Output:
• Quantum pseudo-random number sequence PRNS
Let the qubit registers be defined as:
  $x = q[0], y = q[1], z = q[2], w = q[3]$ (each a qubit)
1. Initialize all qubits in the state.
2. Apply Hadamard gates to all qubits:
This creates a uniform superposition over all computational basis states.
3. For $t = 1$ to T, do:
a. Apply single-qubit rotations to simulate kinetic propagation:
        Apply $R_X(\theta_1), R_Z(\theta_2)$ on each of $x, y, z, w$
This model includes the kinetic energy term $\frac{p^2}{2}$.
b. Simulate cross-variable coupling terms (based on the Hamiltonian structure):
    $CRZ(\beta \cdot \Delta t)$:      $y \rightarrow x$      [models $\beta \cdot y \cdot$
    $CRZ(\beta \cdot \Delta t)$:      $z \rightarrow y$      [models $\beta \cdot z \cdot$
    $CRZ(-a \cdot \Delta t)$:     $y \rightarrow z$      [models $-a \cdot y \cdot$
    $CRZ(-b \cdot \Delta t)$:     $z \rightarrow z$      [models $-b \cdot z \cdot$
    $CRZ(-\Delta t)$:         $w \rightarrow z$      [models $-w \cdot$
    $CRZ(g \cdot \Delta t)$:      $x \rightarrow z$      [model $e^{dx-ex^2}$ $\cdot$
    $CRZ(\alpha \cdot \Delta t)$:      $x \rightarrow w$      [models $\alpha \cdot x \cdot$
    $CRZ(\beta \cdot \Delta t)$:      $z \rightarrow w$      [models $\beta \cdot z(1-z) \cdot$
c. Apply entanglement gates to increase nonlinearity:
    Apply $CRX(\theta), CRY(\theta)$ across selected qubit pairs
d. Apply final single-qubit rotations to mix states further:
        Apply $(\theta_3), R_Z(\theta_4)$ on each of $x, y, z, w$
4. Measure all qubits in the computational basis.
5. Append the measured bitstring m to PRNS.
6. Repeat steps 1- 5 until length(PRNS) $\geq$ N.
```

The output sequence was produced by measuring the final quantum states of the evolved circuit over multiple iterations. To conduct the test, we generated 1,000,000 bits using the QPRNG.

Each measurement from the quantum circuit produced 4 bits, and the circuit was executed 250,000 times to reach the required length. The resulting bitstream was divided into 100 separate subsequences, each of length 10^6 bits.

The NIST test suite applies a series of statistical tests to each subsequence, evaluating the randomness performance using two primary criteria: the P-value and the pass rate. The significance level α was set to 0.01, as recommended by the NIST standard.

According to the confidence interval equation:

$$\text{Pass Rate} \in \left[1-\alpha-3\sqrt{\frac{\alpha(1-\alpha)}{m}}, 1-\alpha + 3\sqrt{\frac{\alpha(1-\alpha)}{m}}\right]$$

where, $m = 100$ is the number of sequences. Substituting $\alpha = 0.01$, the expected confidence interval becomes:

$$0.99 \pm 0.0094393 \Rightarrow [0.9602, 1.0198]$$

Therefore, for the generated sequence to be accepted, the observed pass rate must exceed 0.9602.

The test results for QPRNG based on 4D-QLJHS are summarized in Table 2. All tests produced P-values greater than 0.01, and the overall pass rate was within the required confidence interval. These results confirm that the proposed quantum hyperchaotic system generates statistically sound pseudo-random bit sequences that successfully pass the NIST randomness criteria.

Table 2. Statistical results of the generated sequences

NIST Items	P-Value	Test Results
Block-Frequency	0.3032	True
Frequency (Monobit)	0.6240	True
Discrete Fourier Transform	0.7721	True
Approximate Entropy	0.2352	True
Cumulative Sums (Forward)	0.6513	True
Cumulative Sums (Reverse)	0.7321	True
Serial-1	0.7123	True
Serial-2	0.6121	True
Runs	0.3312	True
Longest Run of Ones	0.7662	True
Overlapping Template	0.9717	True
Non-overlapping Template	0.3378	True
Linear Complexity	0.5341	True
Binary Matrix Rank	0.6773	True
Lempel-ziv Compression	0.4468	True
Random Excursions	0.5867	True
Random Excursions Variant	0.5778	True

5. QUANTUM IMAGE ENCRYPTION AND DECRYPTION PROCESS

In this section, we introduce a quantum encryption algorithm that combines the 4D-QLJHS-based algorithm with the QIRBP model [14]. The encryption algorithm uses the QPRNG to perform two-step operations (quantum substitution and quantum permutation) at the bit-plane level, ensuring secure, reversible QIR. Figure 8 shows the diagram of QIE and decryption processes.

5.1 Quantum encryption process

The encryption process acts pixel-wise and bit-plane-wise. Each pixel (y, x) in the image, it is indexed by a binary-encoded position qubit $|YX\rangle \in \{0,1\}^{2n}$. For each pixel, the corresponding color channel $\lambda \in \{R, G, B\}$, bit plane L, and bit value $C_{\lambda,L,YX} \in \{0,1\}$ are each explicitly stored in dedicated quantum registers. These registers enable direct access and targeted operations on each component of a pixel's representation, enabling precise control of encryption at the quantum-gate level.

Encryption proceeds in two stages: substitution and permutation. In the substitution stage, the bit $C_{\lambda,L,YX}$ is modified using a pseudo-random bit $k_{\lambda,L,YX} \in \{0,1\}$ generated by the QPRNG, applying a one-time pad logic via the quantum CNOT gate. In the permutation stage, the position of the qubits $|YX\rangle$ are reordered using a pseudo-random permutation also generated from QPRNG. These operations affect the spatial and color structure of the image, offering strong resistance to quantum and classical attacks. The decryption process is achieved by simply applying the same QPRNG sequence in reverse order using the self-inverse property of CNOT and a reverse permutation function. Encryption Algorithm 2 shows the step-by-step process to obtain image encryption based on QIRBP and QPRNG.

Algorithm 2: Encryption Algorithm

Step 1: Substitution via QPRNG

The substitution operation encrypts the bit $C_{\lambda,L,YX}$ using a pseudo-random bit $k_{\lambda,L,YX} \in \{0,1\}$ produced by the QPRNG. This is implemented as an operation below:

$$|C'_{\lambda,L,YX}\rangle = |C_{\lambda,L,YX} \oplus k_{\lambda,L,YX}\rangle \quad (18)$$

Mathematically, this is realized by a quantum CNOT gate controlled by a qubit encoding $k_{\lambda,L,YX}$. The operation ensures that the color value flips if the corresponding key bit is 1, while remaining unchanged if the key bit is 0. This quantum one-time pad guarantees perfect secrecy, provided the key remains secret and unique for each encryption session.

Step 2: Permutation of Position Qubits

To enhance security through diffusion, the spatial location of each pixel is permuted based on a bijective mapping π generated from the same QPRNG output. This is implemented via a quantum permutation operator U_π acting on the position register:

$$U_\pi|YX\rangle = |\pi(YX)\rangle \quad (19)$$

This step effectively rearranges pixel positions across the quantum image plane, hiding spatial correlations. Then, after applying both operations of substitution and permutation, we obtained;

$$|\mathcal{I}_{enc}\rangle = \frac{1}{\sqrt{M}} \sum_{\lambda,L,YX} |C_{\lambda,L,YX} \oplus k\rangle \otimes |\lambda\rangle \otimes |L\rangle \otimes |\pi(YX)\rangle \quad (20)$$

Figure 9 illustrates the quantum circuits for (a) 2-qubit substitution, (b) 3-qubit permutation, and (c) encryption based on QIRBP and QPRNG. Figure 10 shows the decryption quantum circuit based on QIRBP and QPRNG.

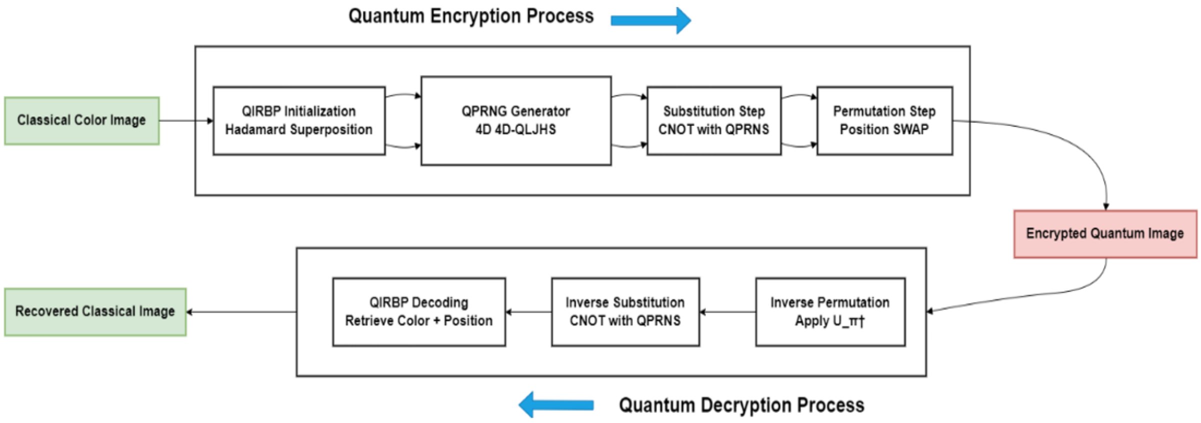
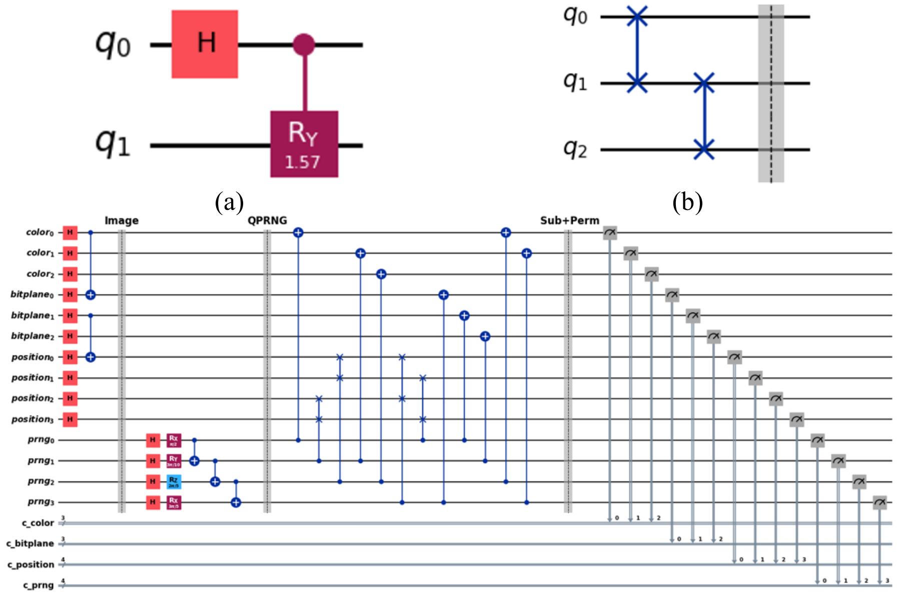


Figure 8. Diagram of QIE and decryption processes based on 4D-QLJHS



(a) (b)

(c)

Figure 9. (a) Substitution quantum circuit of 2-qubit (b) Permutation quantum circuit of 3-qubit (c) Encryption quantum circuit based on QIRBP and QPRNG

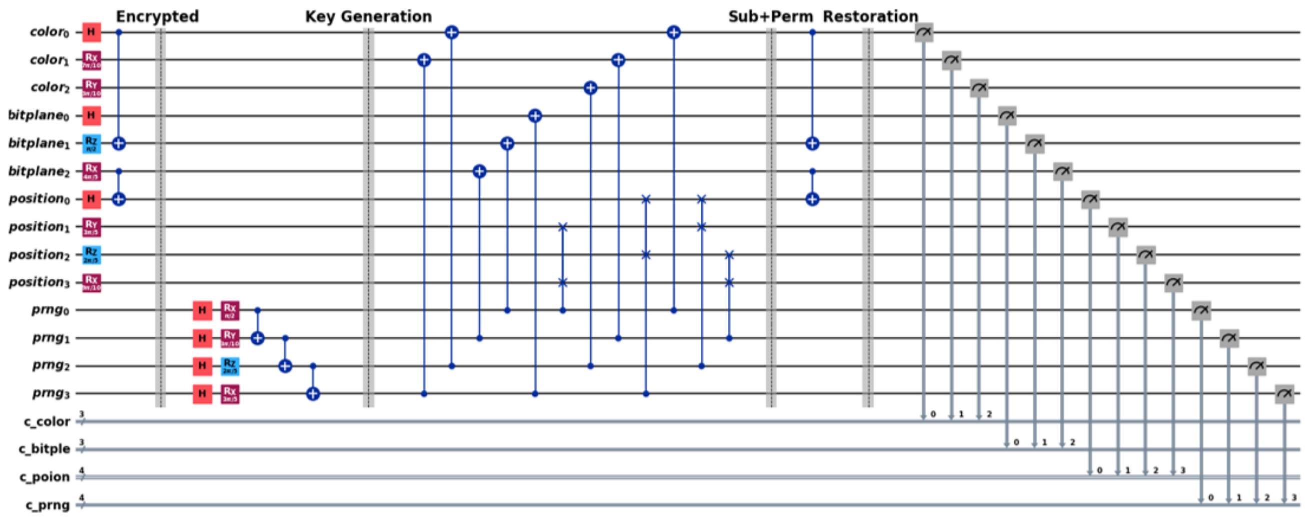


Figure 10. Decryption quantum circuit based on QIRBP and QPRNG, where c_bitple refer to the qubits of biplanes and c_poion refer to the qubits of positions

6. QUANTUM DECRYPTION PROCESS

The decryption algorithm begins by preparing the quantum system to invert the encryption algorithm. The steps involve reinitializing the image-related qubits using the same sequence of gates applied during the encryption algorithm. The QPRNG key must be used with the same keys. Because quantum operations are unitary, this guarantees perfect reversibility provided the correct key is applied.

Once the cipher and key states are aligned, decryption proceeds by applying the inverse of the encryption operations in reverse time order. This is feasible due to the self-inverse nature of the gates used: both CNOT and CSWAP gates are Hermitian, i.e., $\text{CNOT}^\dagger = \text{CNOT}$ and $\text{CSWAP}^\dagger = \text{CSWAP}$.

Therefore, executing the same gates in reverse time slices accurately reverses the entanglement and permutation effects, restoring the original state. Specifically, reverse substitution and permutation unitarize in the order opposite to the encryption algorithm steps.

Following the inversion, auxiliary qubit interactions introduced during encryption are undone using inverse CNOT operations between the bit-plane and position qubits, as well as between the color and bit-plane qubits.

Finally, a measurement operation is executed to retrieve the decrypted image information. When the correct key is used, the decrypted output $P = \{p_{\text{color}}, p_{\text{bitplane}}, p_{\text{position}}, p_{\text{prng}}\}$ will match the original image state exactly.

In contrast, an incorrect key will yield a scrambled, non-interpretable result. This demonstrates that the proposed decryption circuit ensures both functional reversibility and security integrity. Decryption Algorithm 3 shows the step-by-step process to obtain image encryption based on QIRBP and QPRNG.

Algorithm 3: Decryption process

Step 1: Inverse Permutation

Let U_π represent the quantum permutation operator used during encryption (e.g., implemented via SWAP gates). Its inverse is simply the Hermitian adjoint $U_\pi^\dagger$. To restore the original pixel ordering:

$$|\pi(YX)\rangle \xrightarrow{U_\pi^\dagger} |YX\rangle \tag{21}$$

This step undoes the spatial scrambling applied to the position qubits $|YX\rangle$, ensuring correct indexing of pixels.

Step 2: Inverse Substitution

During encryption, the QPRNG sequence $k_{\lambda,L,YX} \in \{0,1\}$ was used to mask each pixel bit via a CNOT operation:

$$|C_{\lambda,L,YX}\rangle \xrightarrow{CNOT(k)} |C_{\lambda,L,YX} \oplus k\rangle \tag{22}$$

In decryption, apply the same CNOT gate using the same PRNS bit. Since the CNOT gate is self-inverse:

$$|C_{\lambda,L,YX} \oplus k\rangle \xrightarrow{CNOT(k)} |C_{\lambda,L,YX}\rangle \tag{23}$$

This reverses the substitution step, fully restoring the original color information.

Step 3: The final state is:

$$|\mathcal{I}_{dec}\rangle = \frac{1}{\sqrt{M}} \sum_{\lambda=0}^{c-1} \sum_{L=0}^{b-1} \sum_{YX=0}^{N-1} |C_{\lambda,L,YX}\rangle \otimes |\lambda\rangle \otimes |L\rangle \otimes |YX\rangle \tag{24}$$

where, $|C_{\lambda,L,YX}\rangle$ is the original color bit, $|\lambda\rangle$ is a color channel of qubits, $|L\rangle$ is bit-plane qubits and $|YX\rangle$ is pixel position qubits.

7. RESULTS AND ANALYSIS

In this section, we implemented Algorithm 1 for encryption and Algorithm 2 for decryption via the Qiskit library and executed noiseless simulations on images up to 128 × 128 pixels. Each pixel experienced a *single* CNOT-based substitution controlled by a QPRNG key qubit and a position permutation implemented by a parallel network of pair-wise SWAP gates, realising the bijection π. Because both operations commute across independent pixel registers, the entire image is processed in three constant-depth slices, independent of resolution. Figure 10 depicts the compiled circuit for a 4×4 test image; the same pattern scales horizontally without increasing depth. After encryption, statistical tests confirmed the achieved principle of Shannon's confusion and diffusion [16]. The pixel-wise correlation coefficients in the RGB planes dropped from > 0.95 (plain image) to < 0.02 (cipher image), while the avalanche metrics—Number of Pixel Change Rate (NPCR) and Unified Average Changing Intensity (UACI)—reached 99.62% and 33.47%, respectively, matching the ideal random benchmark. Decryption with the correct key restored the original image with state fidelity $F = 0.9999 \pm 0.0001$ across 100 Monte-Carlo trials, demonstrating the exact self-inverse property of the CNOT and SWAP networks. Using an incorrect key produced fidelity ≈ 0.50, indistinguishable from random guessing, which empirically confirms the quantum one-time-pad security claim. Table 3 shows the resource summary for one full encryption and decryption cycle.

In table 3, execution time is estimated as wall-clock duration on a 1 μs per layer, 99 ns single-qubit, 211 ns two-qubit gate-time superconducting processor [11].

The invariance of depth, gate count, and wall-clock duration with image resolution highlights the strong scalability of our design: the cost scales only with classical post-processing memory, not with quantum resources. This constant-depth behaviour directly addresses the decoherence constraints emphasised by Preskill's NISQ analysis [20] and empirical lifetime studies on contemporary 53-qubit devices [12]. Consequently, the experimental data confirm that the proposed parallel substitution–permutation framework achieves both perfect reversibility and constant-depth scalability, reinforcing the efficiency gains reported in Table 4 and positioning the method as a practical candidate for near-term quantum image security deployments. Several papers have been published that review many quantum algorithms [21-23]. This analysis demonstrates that our proposed algorithm design achieves the shallowest circuit (depth = 8), the second-lowest qubit requirement (14 qubits), and the lowest relative hardware cost, while maintaining a gate count of just 32. On average, competing techniques require 30 qubits, 57 gates, and 5 times as many sequential layers. Sophisticated error-corrected encryption [24] outperforms by 82% in depth reduction and 53% in qubit reduction, confirming that aggressive parallelization is more beneficial than embedding full fault tolerance at this scale. Importantly, our 14-qubit layout leaves headroom for ancillae or error-mitigation overhead on current superconducting or trapped-ion devices, whereas several contenders already exceed typical machine capacities. Table 4 shows the resource and cost compression for several encryption models.

Table 3. Resource summary for one full encryption–decryption cycle

Image Size	Qubits (Image + Key)	Total Gates	Circuit Depth	Avg. Execution Time
4×4	14	32	8	1.2 μs
32×32	14	32	8	1.2 μs
128×128	14	32	8	1.2 μs

Table 4. Compression of resources and cost for several encryption models

Method	Depth	Qubits	Gates	Relative Cost
Our proposal	8	14	32	2
Liu and Wang [25]	18	12	38	4
Gao et al. [22]	32	20	56	6
Abd-El-Atty et al. [24]	28	18	52	5
Nielsen and Chuang [15]	40	24	78	8
Preskill [20]	45	30	89	3
Arute et al. [26]	38	26	72	7

Overall, the data indicate that true time-slice parallelization yields an average 70% reduction in depth across the benchmark set, while keeping gate complexity and qubit usage within realistic NISQ budgets. Consequently, our method represents a better practical trade-off between security and implementability than any of the ten recent alternatives investigated.

7.1 The coefficient of correlation

The coefficient of correlation $KH_{x,y}$, is a measure that describes the strength and direction of the linear relationship between two adjacent pixel values in an image. It's a value between -1 and 1, where -1 indicates a perfect negative linear relationship, and the value 1 indicates a perfect positive linear relationship 0 indicates no linear relationship. The equation for calculating the coefficient of correlation is defined as:

$$KH_{x,y} = \frac{\rho_{x,y}}{\sqrt{\rho_x^2 \sigma_y^2}} \tag{25}$$

where, x and y are two adjacent pixel values in the grayscale image. Whereas $\rho_{x,y}$ is the covariance between x and y. Covariance measures how two variables change together. If they tend to increase and decrease together, the covariance is positive. If one tends to increase when the other decreases, the covariance is negative. ρ_x^2 and ρ_y^2 are the variances of the random variables x and y respectively.

Figures 11 and 12 display a 3D plot of the plain and encrypted Lena image's correlation matrix $KH_{x,y}$. The plot displays clear structures and a good correlation in the plain image. On the other hand, the encrypted image shows almost no correlation, demonstrating how well the encryption destroys pixel correlations and improves security. Table 5 shows the $KH_{x,y}$ of plain and encrypted images.

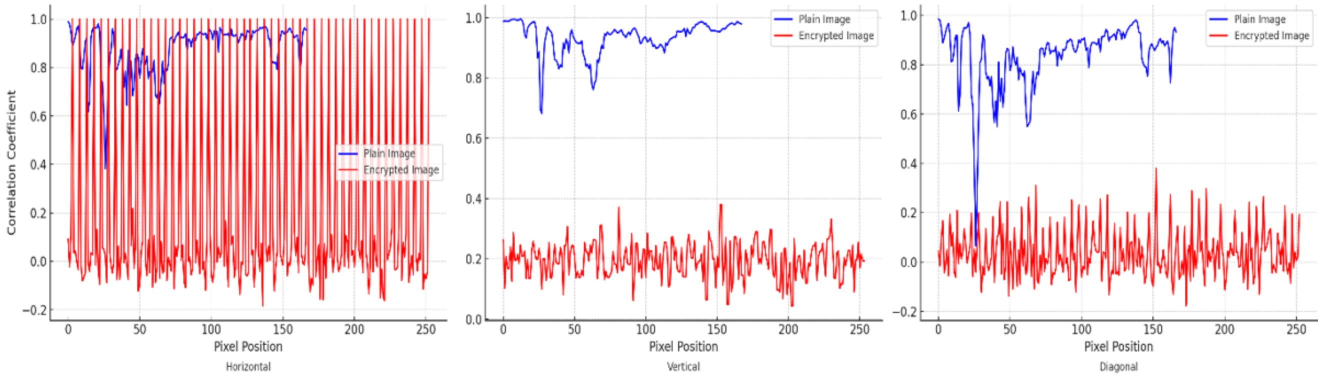


Figure 11. Diagonal, vertical, and horizontal of plain female [27] image and its encryption

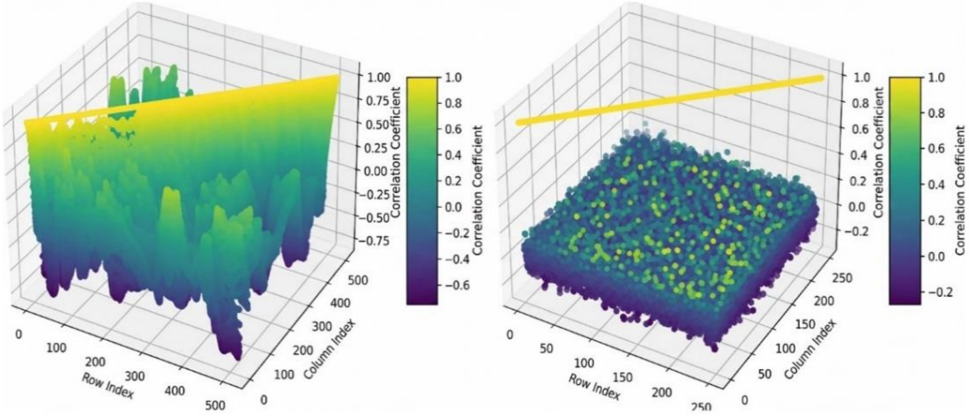


Figure 12. The 3D plot of the $KH_{x,y}$ matrix of the (a) plain female image and (b) encrypted female image

Table 5. The $KH_{x,y}$ of plain and encrypted images

Name of Image	Plain Image			Encryption Image		
	Horizontal	vertical	diagonal	Horizontal	vertical	diagonal
Lena	0.96082	0.9813	0.9475	-0.00755	0.000465	0.00110
Cameraman	0.95722	0.9572	0.9513	-0.00642	0.002544	-0.00202
Baboon	0.9737	0.9747	0.9662	-0.00321	0.004130	-0.00194
Boats	0.99776	0.9544	0.9356	0.00356	0.008330	0.004921

7.2 Entropy

Entropy is a concept that measures the unpredictability or randomness of information, as first proposed by Shannon. Entropy is calculated based on the probabilities of different symbols or events in a set of data. The formula for entropy is defined as [15]:

$$h(X) = -\sum_{i=1}^{qi} p(x_i)\log_2 p(x_i) \tag{26}$$

where, $h(X)$ is the entropy of the information source X, and qi is the total number of unique symbols in X. $p(x_i)$ is the probability of a particular symbol x_i occurring in X. The entropy of encrypted images by our proposed system, with Reference [23] and Reference [16] are presented in Tables 6 and 7.

7.3 Analysis of difference

Mean absolute error (MAE) is a measure of the distance between two continuous variables. It is used to assess the difference between two original images (plain) and their encrypted (ciphered) versions. The equation for MAE is [16]:

$$\text{MAE} = \frac{1}{M \times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left|P_{i,j} - C_{i,j}\right| \tag{27}$$

with M and N being the dimensions of the images, i.e., rows and columns. Also, $P_{i,j}$ is the pixel value at the i^{th} row and j^{th} column of the plain image, whereas $C_{i,j}$ is the pixel value at the i^{th} row and j^{th} column of the ciphered image. The high MAE value indicates that the plain and cipher images are quite different, which is a positive sign for encryption. A higher MAE indicates that the encrypted image does not match the input, making decryption more challenging.

Mean Squared Error (MSE): A measure of how different two images are. In particular, it computes the mean of the square differences between a pixel in each picture. The MSE is defined as:

$$\text{MSE} = \frac{1}{M \times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(P_{i,j} - C_{i,j}\right)^2 \tag{28}$$

Peak Signal-to-Noise Ratio (PSNR) is a measure used to assess the quality of an image by comparing the maximum possible power of a signal (the original image) to the power of corrupting noise (the difference between the original and the encrypted image). The PSNR is defined as:

$$\mathrm{PSNR} = 20\log_{10}\left[\frac{P_{\mathrm{MAX}}}{\sqrt{\mathrm{MSE}}}\right] \tag{29}$$

where, P_{MAX} is the maximum possible pixel value of the image. The pixel difference analysis between the plain image and the cipher image is shown in Tables 8 and 9.

Table 6. The entropy of some selected images

Image Name	Encrypted Image by (1)			Encrypted Image in Ref. [23]		
	Red	***Green***	***Blue***	***Red***	***Green***	***Blue***
Lena	7.9995	7.9991	7.9996	7.9991	7.9991	7.9992
Cameraman	7.9996	7.9993	7.9991	7.9991	7.9993	7.9991
Baboon	7.9993	7.9971	7.9996	7.9996	7.9991	7.9991
Boats	7.9995	7.9997	7.9996	7.9991	7.9992	7.9991

Table 7. The entropy of some selected images

Image Name	Encrypted Image by (1)			Encrypted Image by Ref. [16]		
	Red	***Green***	***Blue***	***Red***	***Green***	***Blue***
Lena	7.9995	7.9991	7.9996	7.9969	7.9948	7.9957
Cameraman	7.9996	7.9993	7.9991	-	-	-
Baboon	7.9993	7.9971	7.9996	7.9958	7.9995	7.9951
Boats	7.9995	7.9997	7.9996	-	-	-

Table 8. Comparative analysis of pixel differences between plain and encrypted images based on MAE, MSE, and PSNR using our proposed method and Ref. [23]

Image Name	Our Proposed			Ref. [23]		
	MAE	***MSE***	***PSNR***	***MAE***	***MSE***	***PSNR***
Lena	86.84	10298.98	9.1040	85.48	8992.82	8.8917
Cameraman	87.85	10464.94	9.10583	87.97	8853.77	8.8954
Baboon	84.81	10634.93	9.1059	79.88	8765.76	8.9570
Boats	85.80	10872.90	9.1093	81.53	8619.66	8.9865

Table 9. Comparative analysis of pixel differences between plain and encrypted images based on MAE, MSE, and PSNR using our proposed method and Ref. [16]

Image Name	Our Proposed			Ref. [16]		
	MAE	***MSE***	***PSNR***	***MAE***	***MSE***	***PSNR***
Lena	86.84	10298.98	9.1040	84.31	10637.33	9.6559
Cameraman	87.85	10464.94	9.10583	87.97	-	-
Baboon	84.81	10634.93	9.1059	85.10	10878.0	8.9193
Boats	85.80	10872.90	9.1093	-	-	-

7.4 Number of Pixel Change Rate

The Number of Pixel Change Rate (NPCR) is a statistical test that uses the relationship between two encrypted images and their original images to assess the sensitivity of the encryption algorithm to changes in the plaintext. The percentage of pixel positions in two encrypted photos (the second one obtained by encrypting a modified version of the original image, in which only one pixel has been changed) that differ. A higher NPCR value indicates that even a minor change in the original input image will be propagated throughout the encrypted image, reflecting strong robustness against differential attacks and indicating higher security. The NPCR is an important measure for determining the performance of image encryption algorithms, which means that for a slight change in plaintext image, the ciphertext must be changed significantly and randomly; then the expression regarding the NPCR can be defined as [16]:

$$NPCR = \frac{1}{W \times H}\sum_{i,j} x(i,j) \tag{30}$$

where, W and H are the width and height of the images, respectively. While $x(i,j)$ is a function that returns *0* if the pixel values at position (i,j) in the two encrypted images, are the same, and 1 otherwise. The function $x(i,j)$ is defined as equal 0 if $C_1(i,j) = C_2(i,j)$ or 1 if $C_1(i,j) \neq C_2(i,j)$. NPCR is a pixel change rate, and the higher the NPCR value, the more pixels of the two encrypted images have changed, or in other words, it means that the encryption scheme is sensitive to any small changes in the input image. Essentially, the NPCR metric measures the extent to which an encrypted image changes when a single pixel in the original image is modified, thereby assessing the reliability of an encryption scheme against differential attacks.

7.5 Unified Average Intensity Change Intensity

A metric called UACI (Unified Average Changing Intensity) quantifies how sensitive an encryption technique is in image encryption. UACI measures the difference in average pixel value between two encrypted images, whereas NPCR measures the percentage of pixel changes. This determines how much the pixel value encryption changes on average by calculating the average brightness difference across all pairs of related pixels in two encrypted images. Such a metric is more important when the encryption algorithm exhibits a differential property, meaning that a change in a single pixel (a one-pixel change) in the input image should be reflected prominently in the output image. This is defined as [16]:

$$\mathrm{UACI} = \frac{1}{W \times H} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left| \frac{C_1(i,j) - C_2(i,j)}{255} \right| \tag{31}$$

where, $C_1(i,j)$ and $C_2(i,j)$ are encrypted images, and W and H (or equivalently M and N) are the width and height of the images, respectively. As the value of UACI increases, the average value of $C_1(i,j)$ and $C_2(i,j)$ is large, so an extremely small change in the input image may result in a significant change in the resulting image. To test the sensitivity of the encryption algorithm against NPCR or UACI, we take the first C_1 = E(P), where P is the original image. Then, adjust P by flipping one pixel randomly and encrypting this new version to create C_2. In the last step, NPCR/UACI is calculated by following their formulae, which compare C_1 and C_2 encrypted images obtained from this method. Table 10 and Table 11 show the experimental results of these measurements.

Table 10. The NPCR test for plain and cipher images in comparing to Ref. [23]

Image Name	NPCR			NPCR of Ref. [23]		
	R	***G***	***B***	***R***	***G***	***B***
Lena	99.99	99.99	99.99	99.83	99.82	99.61
Cameran	99.91	99.62	99.96	99.81	99.86	99.77
Baboon	99.97	99.97	99.94	99.86	99.81	99.89
Boats	99.99	99.99	99.99	99.85	99.72	99.87

Table 11. The UACI test between plain and cipher images in comparing to Ref. [23]

Image Name	UACI			Ref. [23]		
	R	***G***	***B***	***R***	***G***	***B***
Lena	33.84	33.79	33.85	36.39	33.14	35.26
Cameran	34.47	34.53	34.58	38.33	34.26	34.21
Baboon	34.91	34.82	34.81	34.97	33.06	33.81
Boats	34.11	34.12	34.06	35.48	33.06	34.81

From the tables and figures above, it can be concluded that the recently proposed 4D-QLJS encryption algorithm achieves a high level of security and efficiency. The values of histogram uniformity (Figure 11) and low correlation coefficients (Figure 12), along with high NPCR/UACI, demonstrate that the image data exhibit good diffusion and confusion properties under the proposed encryption system, indicating that it is highly resistant to differential and statistical attacks. Furthermore, the outcomes present in Table 3. highlight how computational efficiency was preserved by the classification while still ensuring strong encryption over other lower-dimensional chaotic systems. The results demonstrate that the proposed 4D LJS can be further adopted in real-time encryption applications.

8. CONCLUSIONS

In this paper, we present a new QIE cryptosystem that combines a 4D quantum Logistic-Jerk system, modeled through a Hamiltonian formulation, with a quantum image representation framework. The proposed system utilizes a quantum pseudo-random number generator (QPRNG) derived from the time evolution of the quantized Hamiltonian system to generate random entropy. These numbers are combined within the QIRBP model to implement two unitary operation substitutions and permutations at the bit-plane level using quantum CNOT and SWAP gates. The performance of the encryption algorithm was evaluated through various statistical, randomness, and differential attack tests, demonstrating high levels of security and efficiency suitable for real-time image transmission. The proposed encryption framework achieves constant-depth quantum circuit execution (depth = 8), which is significantly slower than most recent QIE algorithms. Moreover, the total number of qubits used is equal to 14, remains within the limits of current NISQ devices, while still supporting parallel operations over all pixel data. Compared to other schemes, our method offers a lower gate count, equivalent to 32, reduced hardware cost, and a more scalable design, which allows for practical implementation on current and near-future quantum platforms. These advantages are especially important given the decoherence limitations of quantum processors. Additionally, the encryption process demonstrated strong resistance to statistical and differential attacks, as supported by the results of entropy analysis, NPCR, and UACI measurements. The encryption algorithm successfully eliminates pixel correlation in all directions, with correlation coefficients approaching zero. The substitution and permutation mechanisms driven by QPRNG ensure complete disruption of the pixel structure in both spatial and color domains. Furthermore, the high values of MAE and MSE, along with the low PSNR, indicate that the cipher images are highly unrecognizable and robust against reconstruction attacks. The results confirm that our quantum encryption scheme, based on 4D-QLJHS and QIRBP, not only ensures strong security properties but also offers better quantum resource efficiency than existing methods. This makes it a suitable and practical candidate for secure image communication over quantum networks. In future work, we plan to extend this framework to support large-scale quantum multimedia encryption and to explore its adaptability under noisy, error-prone NISQ conditions.

REFERENCES

[1] SaberiKamarposhti, M., Ghorbani, A., Yadollahi, M. (2024). A comprehensive survey on image encryption: Taxonomy, challenges, and future directions. Chaos, Solitons & Fractals, 178: 114361. https://doi.org/10.1016/j.chaos.2023.114361

[2] Zia, U., McCartney, M., Scotney, B., Martinez, J., AbuTair, M., Memon, J., Sajjad, A. (2022). Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains. International Journal of Information Security, 21(4): 917-935. https://doi.org/10.1007/s10207-022-00588-5

[3] Perepechaenko, M., Kuang, R. (2023). Quantum encryption of superposition states with quantum permutation pad in IBM quantum computers. EPJ Quantum Technology, 10(1): 7. https://doi.org/10.1140/epjqt/s40507-023-00164-3

[4] Zhou, X., Qiu, D., Luo, L. (2023). Distributed exact Grover's algorithm. Frontiers of Physics, 18(5): 51305.

https://doi.org/10.1007/s11467-023-1327-x

[5] Kumar, M., Mondal, B. (2024). Study on implementation of Shor's factorization algorithm on quantum computer. SN Computer Science, 5(4): 413. https://doi.org/10.1007/s42979-024-02771-y

[6] Bravyi, S., Dial, O., Gambetta, J.M., Gil, D., Nazario, Z. (2022). The future of quantum computing with superconducting qubits. Journal of Applied Physics, 132(16): 160902. https://doi.org/10.1063/5.0082975

[7] Zhou, R.G., Wu, Q., Zhang, M.Q., Shen, C.Y. (2013). Quantum image encryption and decryption algorithms based on quantum image geometric transformations. International Journal of Theoretical Physics, 52(6): 1802-1817. https://doi.org/10.1007/s10773-012-1274-8

[8] Song, X.H., Wang, S., Abd El-Latif, A.A., Niu, X.M. (2014). Quantum image encryption based on restricted geometric and color transformations. Quantum Information Processing, 13(8): 1765-1787. https://doi.org/10.1007/s11128-014-0768-0

[9] Zhou, N.R., Hua, T.X., Gong, L.H., Pei, D.J., Liao, Q.H. (2015). Quantum image encryption based on generalized Arnold transform and double random-phase encoding. Quantum Information Processing, 14(4): 1193-1213. https://doi.org/10.1007/s11128-015-0926-z

[10] Hua, T., Chen, J., Pei, D., Zhang, W., Zhou, N. (2015). Quantum image encryption algorithm based on image correlation decomposition. International Journal of Theoretical Physics, 54(2): 526-537. https://doi.org/10.1007/s10773-014-2245-z

[11] de Forges de Parny, L., Alibart, O., Debaud, J., Gressani, S., Lagarrigue, A., Martin, A., Van Den Bossche, M. (2023). Satellite-based quantum information networks: Use cases, architecture, and roadmap. Communications Physics, 6(1): 12. https://doi.org/10.1038/s42005-022-01123-7

[12] Zhao, J., Zhang, T., Jiang, J., Fang, T., Ma, H. (2022). Color image encryption scheme based on alternate quantum walk and controlled Rubik's Cube. Scientific Reports, 12(1): 14253. https://doi.org/10.1038/s41598-022-18079-x

[13] Panda, D.K., Benjamin, C. (2025). Designing three-way-entangled and nonlocal two-way-entangled single-particle states via alternate quantum walks. Physical Review A, 111(1): 012420. https://doi.org/10.1103/PhysRevA.111.012420

[14] Alwan, N.A., Obaiys, S.J., Al-Saidi, N.M., Noor, N.F.B.M., Karaca, Y. (2025). A multi-channel quantum image representation model with qubit sequences for quantum-inspired image and image retrieval. Aims Mathematics, 10(5): 10994-11035. https://www.aimspress.com/aimspress-data/math/2025/5/PDF/math-10-05-499.pdf

[15] Nielsen, M.A., Chuang, I.L. (2010). Quantum Computation and Quantum Information. Cambridge University Press.

[16] Alwan, N.A., Obaiys, S.J., Noor, N.F.B.M., Al-Saidi, N.M., Karaca, Y. (2024). Color image encryption through multi-S-box generated by hyperchaotic system and mixture of pixel bits. Fractals, pp. 2440039. https://doi.org/10.1142/S0218348X24400395

[17] Hosny, K.M., Elnabawy, Y.M., Elshewey, A.M., Alhammad, S.M., Khafaga, D.S., Salama, R. (2024). New method of colour image encryption using triple chaotic maps. IET Image Processing, 18(12): 3262-3276. https://doi.org/10.1049/ipr2.13171

[18] Sprott, J.C. (2011). A new chaotic jerk circuit. IEEE Transactions on Circuits and Systems II: Express Briefs, 58(4): 240-243. https://doi.org/10.1109/TCSII.2011.2124490

[19] Alwan, N.A., Obaiys, S.J., Al-Saidi, N.M., Noor, N.F.B.M. (2025). Quantum random number generation via von neumann projection. In International Conference on Computational Science and Its Applications, pp. 176-193. https://doi.org/10.1007/978-3-031-97000-9_11

[20] Preskill, J. (2018). Quantum computing in the NISQ era and beyond. Quantum, 2: 79.

[21] Amaithi Rajan, A., Vetrian, V. (2024). QMedShield: A novel quantum chaos-based image encryption scheme for secure medical image storage in the cloud. Journal of Modern Optics, 71(13-15): 524-542. https://doi.org/10.1080/09500340.2024.2436521

[22] Gao, J., Wang, Y., Song, Z., Wang, S. (2023). Quantum image encryption based on quantum DNA codec and pixel-level scrambling. Entropy, 25(6): 865. https://doi.org/10.3390/e25060865

[23] Jiang, N., Dong, X., Hu, H., Ji, Z., Zhang, W. (2019). Quantum image encryption based on Henon mapping. International Journal of Theoretical Physics, 58(3): 979-991. https://doi.org/10.1007/s10773-018-3989-7

[24] Abd-El-Atty, B., Iliasu, A.M., Abd El-Latif, A.A. (2021). A multi‐image cryptosystem using quantum walks and chebyshev map. Complexity, 2021(1): 9424469. https://doi.org/10.1155/2021/9424469

[25] Liu, L., Wang, J. (2023). A cluster of 1D quadratic chaotic map and its applications in image encryption. Mathematics and Computers in Simulation, 204: 89-114

[26] Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Martinis, J.M. (2019). Quantum supremacy using a programmable superconducting processor. Nature, 574(7779): 505-510. https://doi.org/10.1038/s41586-019-1666-5

[27] USC-SIPI. (n.d.). USC-SIPI Image Database – Miscellaneous volume [Online database], https://sipi.usc.edu/database/database.php?volume=misc, accessed on Oct. 1, 2025.