

Double quantum images encryption scheme based on chaotic system

She-Xiang Jiang(蒋社想), Yang Li(李杨), Jin Shi(石锦), and Ru Zhang(张茹)

Citation: Chin. Phys. B, 2024, 33 (**4**): 040306. DOI: 10.1088/1674-1056/ad1174
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Dilation, discrimination and Uhlmann's theorem of link products of quantum channels

Qiang Lei(雷强), Liuheng Cao(操刘桁), Asutosh Kumar, and Junde Wu(武俊德)
Chin. Phys. B, 2024, 33 (**3**): 030304. DOI: 10.1088/1674-1056/ad0ccb

Unconventional photon blockade in the two-photon Jaynes-Cummings model with two-frequency cavity drivings and atom driving

Xin Liu(刘欣), Meng-Yu Tian(田梦雨), Xiao-Ning Cui(崔晓宁), and Xin-He Zhang(张馨鹤)
Chin. Phys. B, 2024, 33 (**2**): 020308. DOI: 10.1088/1674-1056/ad0cca

Quantum synchronization with correlated baths

Lei Li(李磊), Chun-Hui Wang(王春辉), Hong-Hao Yin(尹洪浩), Ru-Quan Wang(王如泉), and Wu-Ming Liu(刘伍明)
Chin. Phys. B, 2024, 33 (**2**): 020306. DOI: 10.1088/1674-1056/ad0bef

Characterizing entanglement in non-Hermitian chaotic systems via out-of-time ordered correlators

Kai-Qian Huang(黄恺芊), Wei-Lin Li(李蔚琳), Wen-Lei Zhao(赵文垒), and Zhi Li(李志)
Chin. Phys. B, 2022, 31 (**9**): 090301. DOI: 10.1088/1674-1056/ac67c3

Quantum broadcast communication and authentication protocol with a quantum one-time pad

Chang Yan (昌燕), Xu Chun-Xiang (许春香), Zhang Shi-Bin (张仕斌), Yan Li-Li (闫丽丽)
Chin. Phys. B, 2014, 23 (**1**): 010305. DOI: 10.1088/1674-1056/23/1/010305

Double quantum images encryption scheme based on chaotic system

She-Xiang Jiang(蒋社想)[1,2], Yang Li(李杨)[1,†], Jin Shi(石锦)[1], and Ru Zhang(张茹)[1]

[1] *School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan* 232001, *China*
[2] *Anhui Key Laboratory of Mine Intelligent Equipment and Technology, Anhui University of Science & Technology, Huainan* 232001, *China*

(Received 9 September 2023; revised manuscript received 24 November 2023; accepted manuscript online 1 December 2023)

This paper explores a double quantum images representation (DNEQR) model that allows for simultaneous storage of two digital images in a quantum superposition state. Additionally, a new type of two-dimensional hyperchaotic system based on sine and logistic maps is investigated, offering a wider parameter space and better chaotic behavior compared to the sine and logistic maps. Based on the DNEQR model and the hyperchaotic system, a double quantum images encryption algorithm is proposed. Firstly, two classical plaintext images are transformed into quantum states using the DNEQR model. Then, the proposed hyperchaotic system is employed to iteratively generate pseudo-random sequences. These chaotic sequences are utilized to perform pixel value and position operations on the quantum image, resulting in changes to both pixel values and positions. Finally, the ciphertext image can be obtained by qubit-level diffusion using two XOR operations between the position-permutated image and the pseudo-random sequences. The corresponding quantum circuits are also given. Experimental results demonstrate that the proposed scheme ensures the security of the images during transmission, improves the encryption efficiency, and enhances anti-interference and anti-attack capabilities.

Keywords: double quantum images encryption, chaotic system, pixel scrambling, XOR operation

PACS: 03.67.–a, 03.67.Ac, 03.67.Dd, 07.05.Pj **DOI:** 10.1088/1674-1056/ad1174

1. Introduction

With the rapid development of information technology, images have become the most widely used medium for information transmission. Compared to textual information, image data is visual, fast, and cross-linguistic, thus offering a more comprehensive content.[1,2] Encryption is essential in the transmission of images, which can be decrypted by the receiver using the key, thus ensuring the security of the image transmission. Traditional text encryption methods are not suitable for images due to their large data volume, strong data correlation, and high redundancy. Therefore, to protect the security of images during transmission, researchers have proposed various image encryption algorithms.[3–6]

In recent years, breakthroughs have been made in quantum information technology. Combining quantum information theory with image encryption, many effective quantum image encryption methods have been proposed.[7–11] Compared with traditional image encryption techniques, quantum image encryption significantly improves the security and reliability of image encryption due to the advantages of parallelism, efficiency, and security of quantum computing.

The initial task in quantum image processing is to convert classical images into their quantum counterparts.[12] Researchers have designed various quantum image representation models. Le *et al.*[13] proposed a flexible representation of quantum images (FRQI) model, which stored a grayscale image of $2^n \times 2^n$ with a superposition state of $2n+1$ qubits. Zhang *et al.*[14] presented a novel enhanced quantum representation (NEQR) model, which utilized an entangled quantum sequence to indicate the grayscale information accurately. In addition, there are some other models of quantum image representation,[15] such as the normal arbitrary quantum superposition state (NAQSS) model,[16] generalized quantum image representation (GQIR) model,[17] multi-channel quantum image (MCQI) model,[18] quantum image representation based on bit-planes (BRQI) model,[19] and so on.

Based on the introduced representation models, many quantum image encryption algorithms have been proposed.[20–24] They can be broadly classified into two categories, i.e., spatial domain and frequency domain. Spatial domain techniques operate directly on the pixels of the image. Zhou *et al.*[25] presented a quantum image encryption algorithm by performing geometric transformations. Based on the FRQI model, Hou *et al.*[26] proposed an effective quantum image scrambling algorithm. Gao *et al.*[27] proposed an efficient quantum image encryption algorithm by applying quantum DNA coding and pixel-level scrambling. Meanwhile, several quantum image encryption algorithms implemented in the frequency domain have also been proposed. Fan *et al.*[28] accomplished a quantum image encryption for the FRQI model by combining the block-based geometric transformation with Haar wavelet transform. Yang[29] achieved a novel grayscale image encryption and decryption scheme by using quantum Fourier transform and a double random-phase encoding technique. In order to achieve secure and effective encryption results, Gong *et al.*[30] investigated a new single-

†Corresponding author. E-mail: ly1000xyz@163.com

© 2024 Chinese Physical Society and IOP Publishing Ltd

channel quantum color image encryption algorithm based on a hue-saturation-intensity (HSI) model and quantum Fourier transform. Li *et al.*[31] proposed a quantum image encryption algorithm, which used quantum geometric transform, phase-shift transform, and quantum Haar wavelet packet transform to achieve desirable results. Moreover, researchers have also explored quantum multiple-image encryption algorithms to enhance efficiency. Wang *et al.*[32] investigated a double quantum color images representation model (DQRCI) which can store two color digital images simultaneously into a quantum superposition state and obtain a high encryption effect. Liu *et al.*[33] introduced a novel double quantum images encryption approach based on Arnold transform and qubit random rotation. Due to the characteristics of chaotic systems such as randomness, determinism, ergodicity, and sensitivity to initial values, the use of chaos theory for encryption schemes can significantly increase the key space and improve the randomness of encryption. Chaotic systems have been applied to image encryption algorithms by many scholars. Jiang *et al.*[34] utilized a two-dimensional Henon chaotic mapping to encrypt quantum images, which broke away from the restriction of classical computers and worked in quantum computers end to end. According to the higher security of a higher-dimensional chaotic system, a scheme of quantum image encryption and decryption based on a Lorenz hyperchaotic system was introduced by Zhou *et al.*[35] Wang *et al.*[36] proposed an image encryption scheme by using logistic quantum chaos. Jiang *et al.*[37] developed an effective quantum image encryption algorithm by combining a discrete quantum Baker map with a Chen hyperchaotic system, resulting in a huge key space and highly efficient performance. In addition, some cryptography developers have explored quantum image encryption methods that combine DNA coding and chaotic systems.[38,39] During the process of quantum image encryption, it is important to reduce the computational complexity of the algorithm by minimizing the number of qubits involved in the operation. Traditional one-dimensional chaotic systems have a simple structure and a small parameter range, while the high-dimensional hyperchaotic systems have complex structures and require a long time to iteratively generate pseudo-random sequences, making them unsuitable for real-time image transmission.

To address the above-mentioned issues, we design a novel two-dimensional hyperchaotic system based on sine and logistic maps. Simulation results show that the proposed chaotic system exhibits two non-negative Lyapunov exponents, uniformly distributed phase diagrams, and better randomness. The quantum image encryption algorithm proposed in this paper contains the following steps: First, the SHA-256 algorithm is applied to calculate the hash values of the two digital images and then the initial values of the chaotic system are obtained. After that, the two images are transformed into quantum states using the DNEQR model. Second, the chaotic system is iterated with the obtained initial values to generate the pseudo-random sequences. Third, the pixel value scrambling and pixel position permutation are conducted by using the pseudo-random sequences. Finally, the ciphertext image can be obtained by performing forward- and reverse-order XOR diffusion operations between the pixel position-permutated image and pseudo-random sequences, respectively. With the use of a chaotic system in the encryption process, the key space is further increased, and the encryption is more secure and efficient. The main contributions of this paper can be summarized as follows: (1) a double quantum images representation is proposed, which can save more space because it only requires $2n+9$ qubits to store two grayscale images of size $2^n \times 2^n$, while $2\times(2n+8)$ equal to $4n+16$ qubits are needed to prepare the same image information by using the NEQR model. Therefore, the complexity of the calculation is decreased. (2) A new type of two-dimensional hyperchaotic system is proposed, which addresses the disadvantages of narrow parameter space range and obvious periodic window behavior. Moreover, it significantly widens the range of control parameters, solves the drawbacks of slow iteration and computationally extensive of high-dimensional chaotic systems. (3) A chaos-based forward- and reverse-order XOR diffusion method is introduced, which further improves the sensitivity of keys. The original image cannot be recovered even if there is a small change in the keys.

The remaining parts of this paper are organized as follows. Section 2 describes the DNEQR model. In Section 3, the two-dimensional hyperchaotic system along with its performance analysis is introduced in detail. Sections 4 and 5 detail the proposed qubit-level scrambling and pixel position permutation, respectively. In Section 6, the proposed double quantum images encryption and decryption scheme is described in detail. Section 7 shows the simulation experiments and performance analysis. Finally, the conclusion is given in Section 8.

2. The proposed DNEQR model

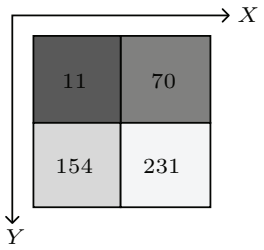
2.1. NEQR model

In 2013, Zhang *et al.*[14] proposed a ground-breaking model called the novel enhanced quantum representation (NEQR). In the model, an image of size $2^n \times 2^n$ and grayscale range $[0, 2^q-1]$, can be expressed as follows:

$$\begin{aligned} |I\rangle &= \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}|f(Y,X)\rangle\otimes|YX\rangle \\ &= \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\bigotimes_{i=0}^{q-1}\left|G_{YX}^{i}\right\rangle\otimes|YX\rangle, \end{aligned} \tag{1}$$

$$\begin{aligned} &f(Y,X) = G_{YX}^{q-1}G_{YX}^{q-2}\cdots G_{YX}^{1}G_{YX}^{0}, \\ &G_{YX}^{i}\in\{0,1\}, \quad f(Y,X)\in[0,2^q-1], \end{aligned} \tag{2}$$

where $f(Y,X)$ denotes the grayscale information in position (Y,X), the symbol $\otimes$ represents the tensor product operation, and the coordinate information in the horizontal and vertical direction is represented by $|YX\rangle = |Y\rangle|X\rangle = |y_{n-1}y_{n-2}\ldots y_0\rangle|x_{n-1}x_{n-2}\ldots x_0\rangle$. Figure 1 shows an example of a 2×2 grayscale image and its NEQR representation.



$$
\begin{aligned}
|I\rangle =& \frac{1}{2}\left(|11\rangle\otimes|00\rangle+|70\rangle\otimes|01\rangle+|154\rangle\otimes|10\rangle+|231\rangle\otimes|11\rangle\right)\\
=& \frac{1}{2}\left(|00001011\rangle\otimes|00\rangle+|01000110\rangle\otimes|01\rangle\right.\\
&\left.+\left(|10011010\rangle\otimes|10\rangle+|11100111\rangle\otimes|11\rangle\right.\right.
\end{aligned}
$$

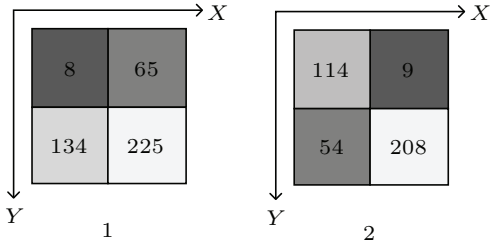
Fig. 1. Example of a 2×2 grayscale image and its NEQR representation.

2.2. Definition of the DNEQR model

Inspired by the NEQR, to fully exploit quantum superposition and quantum entanglement, we propose a double quantum images representation model abbreviated as DNEQR. This model allows for the simultaneous storage of two grayscale images of size $2^n\times 2^n$, which can be represented by the following equation:

$$
\begin{aligned}
|I\rangle &= \frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}|f_1(Y,X)\rangle\otimes|f_2(Y,X)\rangle\otimes|YX\rangle\\
&= \frac{1}{2^n}\sum_{W=0}^{1}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}|P_{WYX}\rangle\otimes|W\rangle\otimes|YX\rangle\\
&= \frac{1}{2^n}\sum_{W=0}^{1}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\left|p_{wyx}^{q-1}p_{wyx}^{q-2}\cdots p_{wyx}^{1}p_{wyx}^{0}\right\rangle\\
&\quad\otimes|W\rangle\otimes|YX\rangle, \qquad (3)
\end{aligned}
$$

where $|YX\rangle$ indicates the coordinate information, and $|f_1(Y,X)\rangle$ and $|f_2(Y,X)\rangle$ denote pixel values of image 1 and image 2, respectively. The grayscale information in position $|YX\rangle$ is represented as $|P_{WYX}\rangle = \left|p_{wyx}^{7}p_{wyx}^{6}\cdots p_{wyx}^{1}p_{wyx}^{0}\right\rangle$ and the range of pixel value is [0, 255]. $|W\rangle$ represents the pixel selection control qubit, $|W\rangle = |0\rangle$ indicates that $|P_{WYX}\rangle$ is the grayscale information of image 1, while $|W\rangle = |1\rangle$ indicates that $|P_{WYX}\rangle$ is the grayscale information of image 2. Two grayscale images of size $2^n\times 2^n$ only require $2n+9$ qubits in DNEQR. For example, two grayscale images of size 2×2 represented as the DNEQR model and the corresponding quantum representation are shown in Fig. 2. The quantum circuit for preparing the DNEQR model of the two images is shown in Fig. 3.



$$
\begin{aligned}
|I\rangle =& \frac{1}{2}\left(|8\rangle\otimes|0\rangle\otimes|00\rangle+|65\rangle\otimes|0\rangle\otimes|01\rangle+|134\rangle\otimes|0\rangle\otimes|10\rangle+|225\rangle\otimes|0\rangle\otimes|11\rangle\right.\\
&+|114\rangle\otimes|1\rangle\otimes|00\rangle+|9\rangle\otimes|1\rangle\otimes|01\rangle+|54\rangle\otimes|1\rangle\otimes|10\rangle+|208\rangle\otimes|1\rangle\otimes|11\rangle)\\
=& \frac{1}{2}\left(|00001000\rangle\otimes|0\rangle\otimes|00\rangle+|01000001\rangle\otimes|0\rangle\otimes|01\rangle+|10000110\rangle\otimes|0\rangle\otimes|10\rangle\right.\\
&+|11100001\rangle\otimes|0\rangle\otimes|11\rangle+|01110010\rangle\otimes|1\rangle\otimes|00\rangle+|00001001\rangle\otimes|1\rangle\otimes|01\rangle\\
&\left.+|00110110\rangle\otimes|1\rangle\otimes|10\rangle+|11010000\rangle\otimes|1\rangle\otimes|11\rangle)\right.
\end{aligned}
$$

Fig. 2. DNEQR model of two grayscale images of size 2×2.

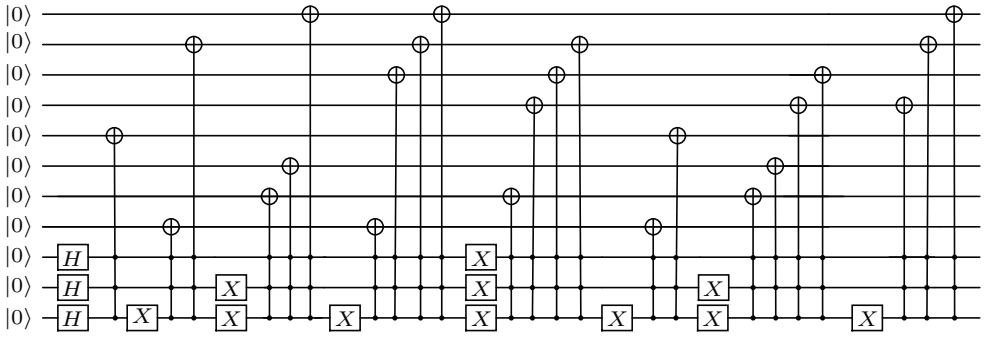


Fig. 3. Preparation of the quantum circuit of two grayscale images of size 2×2 based on the DNEQR model.

3. The proposed two-dimensional hyperchaotic system

3.1. Definition of the two-dimensional hyperchaotic system

Without loss of generality, a two-dimensional chaotic discrete system has the following form:

$$\begin{cases} x_{n+1} = f_1(x_n, y_n), \\ y_{n+1} = f_2(x_n, y_n), \end{cases} \tag{4}$$

where the functions f_1 and f_2 are two chaotic maps, and n is the number of iterations of the system. In this paper, we improve the form of the two-dimensional chaotic discrete system. A novel type of two-dimensional hyperchaotic system is introduced by coupling the sine map and the logistic map together, named 2D-CSLM and defined as

$$\begin{cases} x_{n+1} = \sin^2(\mu \arcsin\sqrt{1-y_n}), \\ y_{n+1} = 1 - \sin(\alpha x_n(1-x_n)\pi), \end{cases} \tag{5}$$

where x_k and y_k are state variables that satisfy x_k, $y_k \in [0,1]$, $k = 0,1,2,n+1$, and μ and α denote the control parameters. When $\alpha = 4$, $\mu \in (0.3, +\infty)$, the chaotic system exhibits perfect chaotic characteristics, entering the chaotic state slightly earlier than indicated in Ref. [40].

3.2. Performance analysis

3.2.1. Bifurcation diagram

The bifurcation diagram can be used to visualize the chaotic behavior of the system. To describe the chaotic properties of the proposed system intuitively, we compare its bifurcation diagram with those of sine map, logistic map, and two other chaotic maps, as shown in Fig. 4.

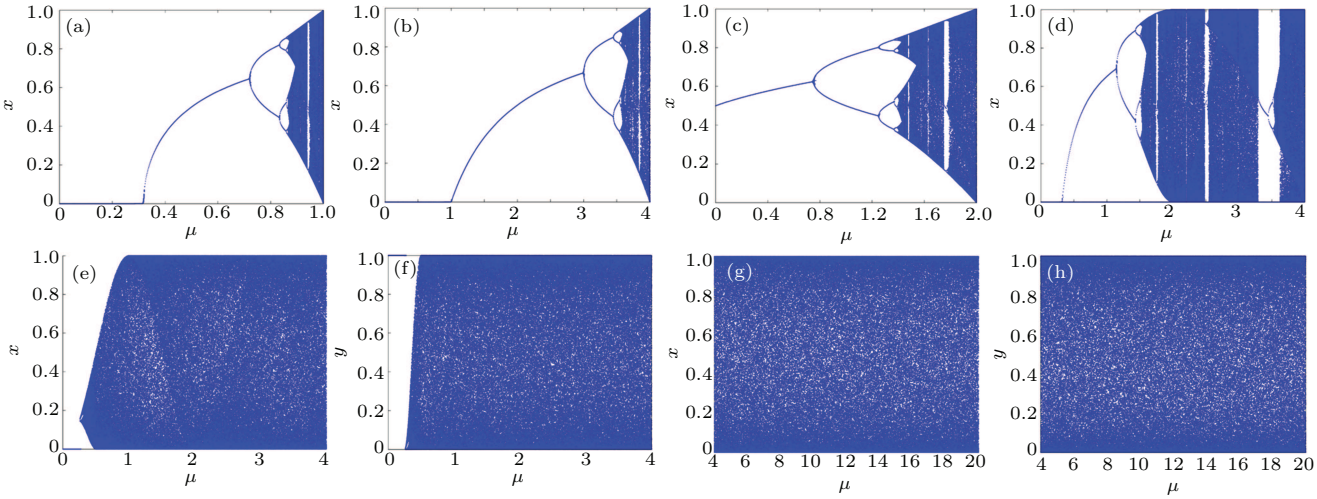


Fig. 4. Bifurcation diagram of (a) sine map, (b) logistic map, (c) quadratic map, (d) sine-logistic map, (e) 2D-CSLM of x sequence for $\mu \in [0,4]$, (f) 2D-CSLM of y sequence for $\mu \in [0,4]$, (g) 2D-CSLM of x sequence for $\mu \in [4,20]$, and (h) 2D-CSLM of y sequence for $\mu \in [4,20]$.

From Figs. 4(a)–4(d), it is obvious that the sine map, logistic map, quadratic map,[41] and sine-logistic map[12] exhibit chaotic properties only in a limited range of parameters and demonstrate a clear periodic window behavior. Figures 4(e) and 4(f) illustrate the bifurcation diagram of the x sequence and y sequence in 2D-CSLM, respectively, when the initial conditions are $x_0 = 0.2185$, $y_0 = 0.7653$ and the control parameters $\alpha = 4$, $\mu \in [0,4]$. As we further increase the value of μ, the system still has outstanding chaotic behavior, which almost presents a full mapping state. Figures 4(g) and 4(h) depict the bifurcation diagram of $\mu \in [4,20]$. As a result, the proposed 2D-CSLM chaotic system demonstrates a wider range of control parameters and exhibits better stochasticity and ergodicity.

3.2.2. Phase diagram

The phase diagram is the typical representation of chaos, showing the distribution of chaotic attractors in a two-dimensional phase plane. A larger region occupied in the phase diagram indicates better chaotic performance of the system. To better illustrate the phase diagram of the system, we select three chaotic systems proposed by others for comparison. The initial conditions are $x_0 = 0.7$, $y_0 = 0.4$, and each chaotic system is iterated 80000 times. Figures 5(a)–5(d) depict the phase diagram of the 2D sine logistic modulation map (2D-SLMM),[42] 2D logistic modulated sine coupling logistic chaotic map (2D-LSMCL),[43] 2D logistic adjusted sine map (2D-LASM),[44] and the proposed 2D-CSLM, respectively. From Fig. 5, it can be observed that the proposed system occupies a larger space compared to the other three maps. Therefore, the pseudo-random sequences generated by our system are more uniform and random.

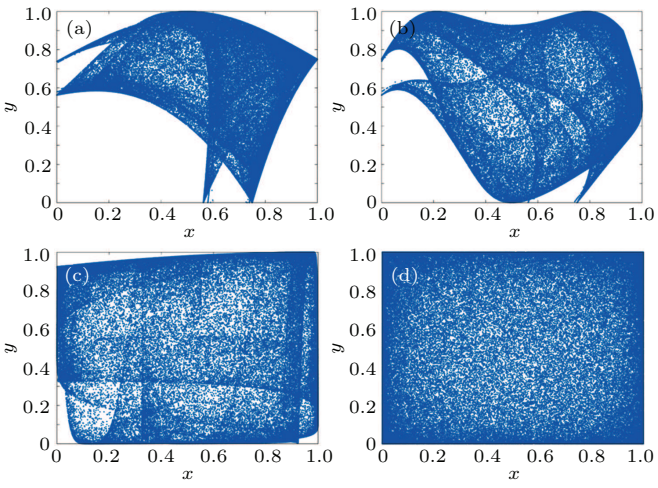


Fig. 5. Phase diagram of (a) 2D-SLMM with $\gamma = 1$, $\eta = 3$, (b) 2D-LSMCL with $\alpha = 0.75$, $\beta = 3$, (c) 2D-LASM with $\mu = 0.5$, and (d) 2D-CSLM with $\alpha = 4$, $\mu = 4.99$.

3.2.3. Lyapunov exponent

The Lyapunov exponent (*LE*) is a numerical feature that represents the average exponential divergence rate of neighboring trajectories in phase space. It is used to assess the stability of dynamic systems. In the case of a one-dimensional dynamic system, its *LE* can be expressed as

$$\lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln\left|f'(x_i)\right|. \tag{6}$$

A positive Lyapunov exponent means that the system is chaotic. A larger *LE* means that the system exhibits better chaotic behavior. Moreover, if a dynamic system has more than one positive *LE*s, it exhibits excellent hyperchaotic behavior. Figure 6 displays the two *LE*s of 2D-SLMM, 2D-LSMCL, and 2D-CSLM, respectively. In Figs. 6(a) and (b), both *LE*s are greater than 0 only within a small parameter range, suggesting the absence of excellent hyperchaotic behaviors. However, as shown in Fig. 6(c), when the parameters of the proposed chaotic system are greater than around 0.3, all *LE*s become positive. Above the curves of Fig. 6(c) is a zoomed-in local area picture. This indicates that the chaotic system exhibits a wider chaotic range and superior hyperchaotic behavior compared to the other two chaotic systems.

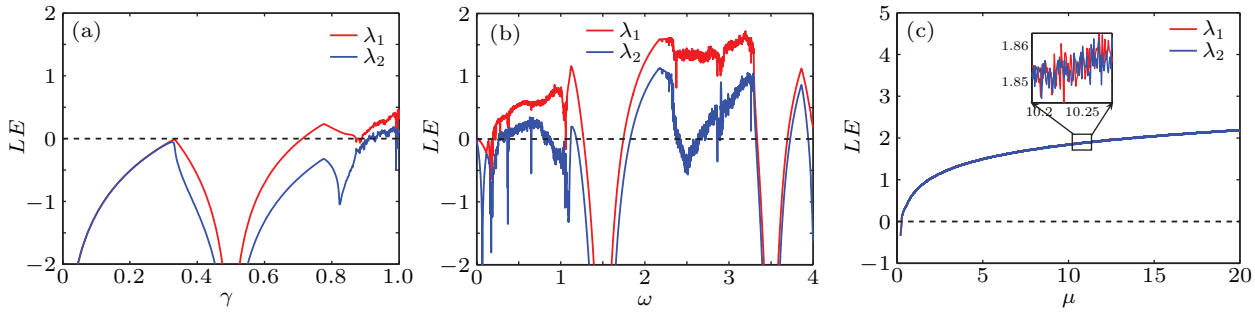


Fig. 6. The *LE*s of (a) 2D-SLMM with $\eta = 3$, (b) 2D-LSMCL with $\beta = 3$, and (c) 2D-CSLM with $\alpha = 4$.

3.2.4. Histogram statistic

Histogram statistics are performed on the sequence generated by the chaotic map to visualize the distribution of values within the chaotic sequence. We vary the control parameters and iteratively generate a pseudo-random sequence of length 30000, then perform histogram statistics on it. The results are presented in Fig. 7. The output sequences of 2D-CSLM display a uniform distribution in the middle part of the range, with more values falling at the ends of the range. On the whole, the sequences generated by 2D-CSLM exhibit a homogeneous distribution within their ranges.

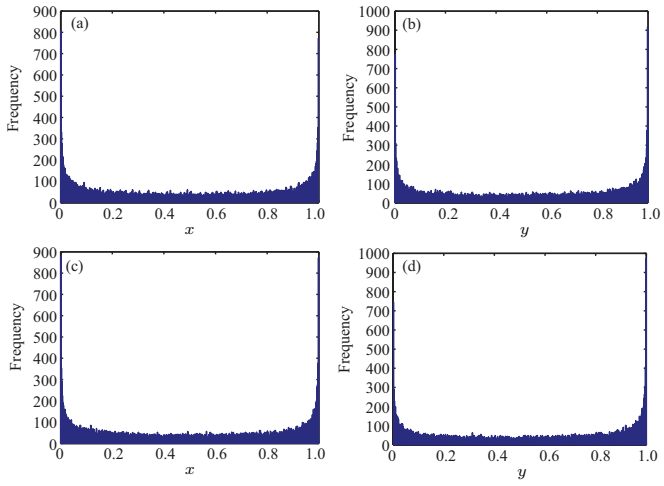


Fig. 7. Histograms of 2D-CSLM with different parameters: (a), (b) $\alpha = 4$, $\mu = 4.99$, (c), (d) $\alpha = 4$, $\mu = 16.99$.

3.2.5. Sensitivity of the initial value

The sensitivity of initial value is a common characteristic of nonlinear systems, and it plays a crucial role in determining the safety of the system. That is, even a slight variation in the initial conditions can lead to completely different output sequences. To assess the sensitivity of initial values of the 2D-CSLM, two sequences are generated using the same control parameters but with a difference of 10^{-15} in initial value. By examining the first 100 values, the output sequence is shown in Fig. 8. The first few values are similar, but after approximately 21 iterations, the output sequences are significantly different. This indicates that the 2D-CSLM is extremely sensitive to initial values. Therefore, the transient effects of chaotic systems can be effectively alleviated.

3.2.6. NIST test

The unpredictability of chaotic sequences is crucial for image encryption.[45] To illustrate the suitability of the proposed chaotic system, we use the NIST SP800-22 test[46] to assess the randomness of the two sequences generated by 2D-CSLM. The SP800-22 test evaluates the randomness of a sequence through 16 sub-tests, each producing a *P*-value; if all *P*-values are greater than 0.01, the test is considered to be passed. Following NIST recommendations, 100 binary streams were tested, with each binary stream 1024000 bits long. Table 1 shows the results of the NIST test, with all *P*-values exceeding 0.01. This indicates that the generated sequences exhibit strong randomness, making them suitable for cryptographic systems.

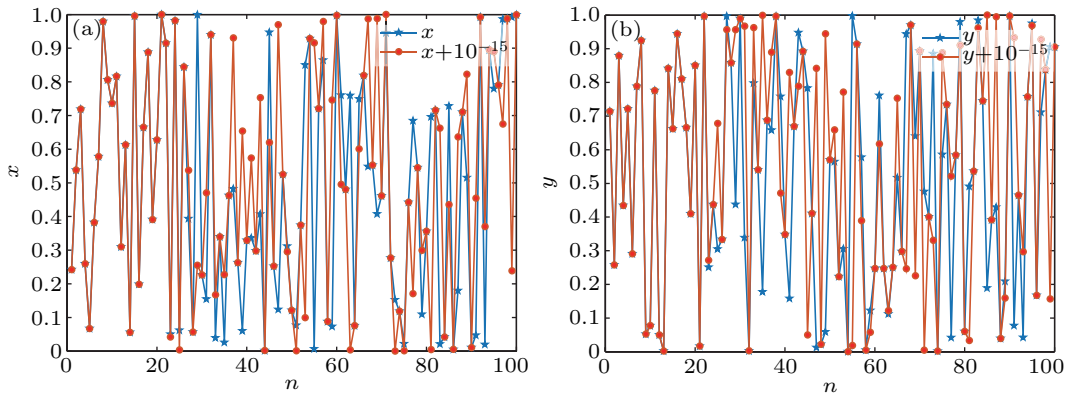


Fig. 8. Sensitivity of the initial value: (a) x sequence of 2D-CSLM, (b) y sequence of 2D-CSLM.

Table 1. Results of the NIST test for 100 sequences generated by 2D-CSLM.

Statistical test	Sequence x			Sequence y		
	P-value	Proportion	State	P-value	Proportion	State
Frequency	0.102526	99/100	Pass	0.213309	98/100	Pass
Block frequency	0.102526	98/100	Pass	0.090936	99/100	Pass
Cumulative sums (forward)	0.616305	98/100	Pass	0.319084	98/100	Pass
Cumulative sums (reverse)	0.115387	99/100	Pass	0.534146	99/100	Pass
Runs	0.224821	97/100	Pass	0.102526	99/100	Pass
Longest run	0.946308	97/100	Pass	0.798139	98/100	Pass
Non-overlapping template	0.666918	100/100	Pass	0.319084	100/100	Pass
Serial	0.883171	100/100	Pass	0.191687	99/100	Pass
Linear complexity	0.637119	99/100	Pass	0.924076	100/100	Pass
Random excursions	0.401199	57/57	Pass	0.568055	62/62	Pass
Random excursions variant	0.867692	57/57	Pass	0.739918	62/62	Pass
Approximate entropy	0.304126	99/100	Pass	0.334538	98/100	Pass
Universal	0.657933	99/100	Pass	0.224821	98/100	Pass
FFT	0.678686	99/100	Pass	0.946308	99/100	Pass
Rank	0.474986	99/100	Pass	0.534146	98/100	Pass
Overlapping template	0.494392	99/100	Pass	0.719747	99/100	Pass

The performance analyses demonstrate that the proposed chaotic system has a wide range of parameters and exhibits hyperchaotic behavior. It also demonstrates favorable chaotic properties and can generate more intricate and random sequences. In terms of application, it is particularly well suited for use in cryptosystems.

4. Qubit-level pixel value scrambling method

In this section, we present a qubit-level random scrambling method, which consists of four pixel-value transformation operations: odd swapping even inverse transformation, even swapping odd inverse transformation, bitwise inverse transformation, and bitwise XOR (exclusion-OR) transformation. The quantum circuits of the four transformations are shown in Figs. 9 and 10, which are composed of quantum swap gates, quantum NOT gates, and quantum controlled-NOT (CNOT) gates. Each of the four transformations is described in detail below.

4.1. Odd swapping even inverse transformation

In general, a quantum grayscale image takes pixel values in the range 0 to 255, and each pixel value can be represented by 8 binary qubits. As shown in Fig. 9(a), we perform swap and inverse operations on the qubits, named odd swapping even inverse transformation and denoted as T_1. That is, first swap the 7th qubit with the 3rd qubit, and the 5th with the 1st, then reverse the 6th, 4th, 2nd, and 0th qubits. The resulting qubit-level can be represented as

$$\begin{aligned} T_1(|I(y,x)\rangle) &= T_1\left(\left|p^7_{wyx}p^6_{wyx}\cdots p^1_{wyx}p^0_{wyx}\right\rangle\right) \\ &= \left|p^3_{wyx}p^{\bar{6}}_{wyx}p^1_{wyx}p^{\bar{4}}_{wyx}p^7_{wyx}p^{\bar{2}}_{wyx}p^5_{wyx}p^{\bar{0}}_{wyx}\right\rangle, \end{aligned} \tag{7}$$

where T_1 can be realized by the quantum swap gates and quantum NOT gates. $|I(y,x)\rangle$ represents the binary form of the pixel value of the image at the coordinate position (y,x). The effect of the quantum swap gate is to interchange the state of two qubits. $\left|p^{\bar{i}}_{wyx}\right\rangle = X\left|p^i_{wyx}\right\rangle$, $i = 6,4,2,0$, where X denotes the quantum NOT gate, which can take the state $|0\rangle$ to the state $|1\rangle$, and vice versa. That is, $X(|0\rangle) = |1\rangle$, $X(|1\rangle) = |0\rangle$. The pixel value changes from $\left|p^7_{wyx}p^6_{wyx}\cdots p^1_{wyx}p^0_{wyx}\right\rangle$ to $\left|p^3_{wyx}p^{\bar{6}}_{wyx}p^1_{wyx}p^{\bar{4}}_{wyx}p^7_{wyx}p^{\bar{2}}_{wyx}p^5_{wyx}p^{\bar{0}}_{wyx}\right\rangle$ by T_1.

4.2. Even swapping odd inverse transformation

Similar to the previous subsection, as shown in Fig. 9(b), swap and inverse operations are performed on the qubits, named even swapping odd inverse transformation and denoted as T_2. That is, first swap the 6th qubit with the 2nd qubit, and

the 4th with the 0th, then reverse the 7th, 5th, 3rd, and 1st qubits. The result is shown below:

$$\begin{aligned} T_2(|I(y,x)\rangle\rangle) &= T_2\left|p^7_{wyx}p^6_{wyx}\cdots p^1_{wyx}p^0_{wyx}\right\rangle \\ &= \left|p^{\bar{7}}_{wyx}p^2_{wyx}p^{\bar{5}}_{wyx}p^0_{wyx}p^{\bar{3}}_{wyx}p^6_{wyx}p^{\bar{1}}_{wyx}p^4_{wyx}\right\rangle. \quad (8) \end{aligned}$$

The pixel value changes from $\left|p^7_{wyx}p^6_{wyx}\cdots p^1_{wyx}p^0_{wyx}\right\rangle$ to $\left|p^{\bar{7}}_{wyx}p^2_{wyx}p^{\bar{5}}_{wyx}p^0_{wyx}p^{\bar{3}}_{wyx}p^6_{wyx}p^{\bar{1}}_{wyx}p^4_{wyx}\right\rangle$ by T_2.

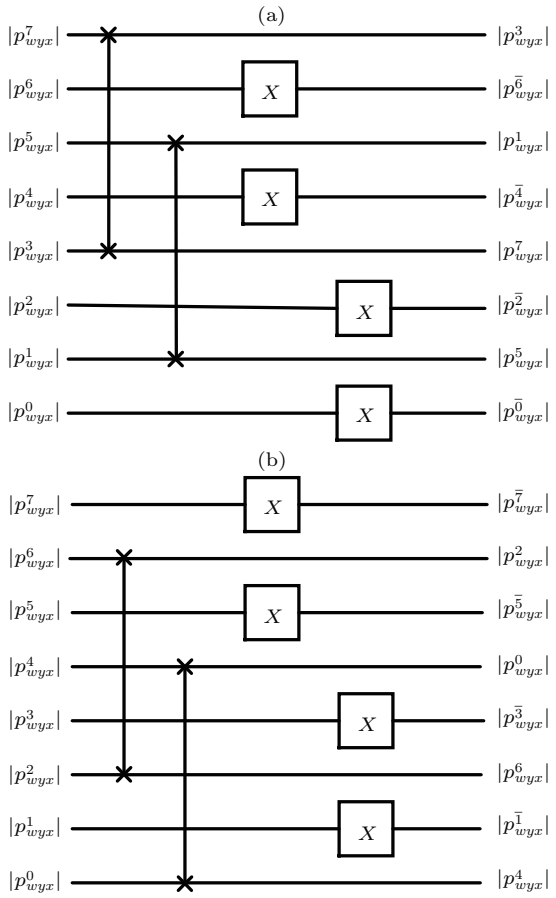


Fig. 9. Quantum circuits for (a) odd swapping even inverse transformation and (b) even swapping odd inverse transformation.

4.3. Bitwise inverse transformation

All 8 binary qubits representing the pixel value are inverted bitwise, denoted as T_3. In Fig. 10(a), the result of this transformation can be expressed as

$$\begin{aligned} T_3(|I(y,x)\rangle\rangle) &= T_3\left|p^7_{wyx}p^6_{wyx}\cdots p^1_{wyx}p^0_{wyx}\right\rangle \\ &= \left|p^{\bar{7}}_{wyx}p^{\bar{6}}_{wyx}\cdots p^{\bar{1}}_{wyx}p^{\bar{0}}_{wyx}\right\rangle. \quad (9) \end{aligned}$$

4.4. Bitwise XOR transformation

As shown in Fig. 10(b), performing bitwise XOR operations on the qubits can further confuse the pixel values of an image. In quantum circuits, CNOT gates are used to implement the XOR operation.[40] The specific implementation process is as follows: First, the 0th qubit is XOR with the 7th qubit to get the new 7th qubit, the 1st qubit is XOR with the 6th qubit to get the new 6th qubit, the 2nd qubit is XOR with the 5th qubit to get the new 5th qubit, and the 3rd qubit is XOR with the 4th qubit to get the new 4th qubit. Then, the new 7th qubit is XOR with the 3rd qubit to obtain the new 3rd qubit, the new 6th qubit is XOR with the 2nd qubit to obtain the new 2nd qubit, the new 5th qubit is XOR with the 1st qubit to obtain the new 1st qubit, and the new 4th qubit is XOR with the 0^{th} qubit to the obtain the new 0th qubit. The qubits obtained are expressed as

$$\begin{aligned} &\left|p^{7'}_{wyx}\right\rangle = \left|p^0_{wyx}\right\rangle \oplus \left|p^7_{wyx}\right\rangle, \quad \left|p^{6'}_{wyx}\right\rangle = \left|p^1_{wyx}\right\rangle \oplus \left|p^6_{wyx}\right\rangle, \\ &\left|p^{5'}_{wyx}\right\rangle = \left|p^2_{wyx}\right\rangle \oplus \left|p^5_{wyx}\right\rangle, \quad \left|p^{4'}_{wyx}\right\rangle = \left|p^3_{wyx}\right\rangle \oplus \left|p^4_{wyx}\right\rangle, \\ &\left|p^{3'}_{wyx}\right\rangle = \left|p^{7'}_{wyx}\right\rangle \oplus \left|p^3_{wyx}\right\rangle = \left|p^0_{wyx}\right\rangle \oplus \left|p^7_{wyx}\right\rangle \oplus \left|p^3_{wyx}\right\rangle, \\ &\left|p^{2'}_{wyx}\right\rangle = \left|p^{6'}_{wyx}\right\rangle \oplus \left|p^2_{wyx}\right\rangle = \left|p^1_{wyx}\right\rangle \oplus \left|p^6_{wyx}\right\rangle \oplus \left|p^2_{wyx}\right\rangle, \\ &\left|p^{1'}_{wyx}\right\rangle = \left|p^{5'}_{wyx}\right\rangle \oplus \left|p^1_{wyx}\right\rangle = \left|p^2_{wyx}\right\rangle \oplus \left|p^5_{wyx}\right\rangle \oplus \left|p^1_{wyx}\right\rangle, \\ &\left|p^{0'}_{wyx}\right\rangle = \left|p^{4'}_{wyx}\right\rangle \oplus \left|p^0_{wyx}\right\rangle = \left|p^3_{wyx}\right\rangle \oplus \left|p^4_{wyx}\right\rangle \oplus \left|p^0_{wyx}\right\rangle, \quad (10) \end{aligned}$$

where the symbol $\oplus$ denotes the quantum XOR operation. The entire process can be expressed as

$$\begin{aligned} T_4(|I(y,x)\rangle\rangle) &= T_4\left(\left|p^7_{wyx}p^6_{wyx}\cdots p^1_{wyx}p^0_{wyx}\right\rangle\right) \\ &= \left|p^{7'}_{wyx}p^{6'}_{wyx}p^{5'}_{wyx}p^{4'}_{wyx}p^{3'}_{wyx}p^{2'}_{wyx}p^{1'}_{wyx}p^{0'}_{wyx}\right\rangle, \quad (11) \end{aligned}$$

where T_4 stands for the bitwise XOR transformation. The pixel value changes from $\left|p^7_{wyx}p^6_{wyx}\cdots p^1_{wyx}p^0_{wyx}\right\rangle$ to $\left|p^{7'}_{wyx}p^{6'}_{wyx}p^{5'}_{wyx}p^{4'}_{wyx}p^{3'}_{wyx}p^{2'}_{wyx}p^{1'}_{wyx}p^{0'}_{wyx}\right\rangle$.

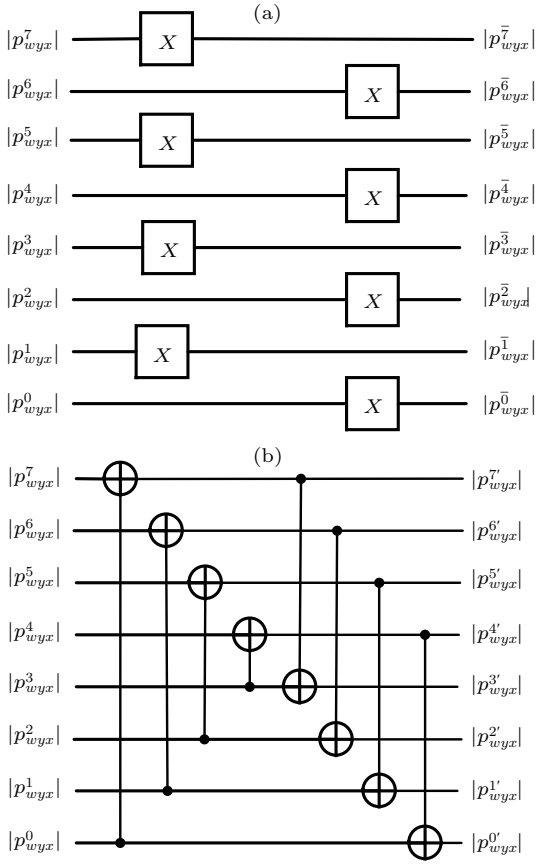


Fig. 10. Quantum circuits for (a) bitwise inverse transformation and (b) bitwise XOR transformation.

The original image will have its pixel values obfuscated by the four transformations described above. To further enhance the randomness of the pixel values, pseudo-random sequences generated by the proposed chaotic system are added to the pixel-value scrambling process, for more details, please refer to Section 6.

5. Pixel position permutation method

In this section, a method for randomly permutating pixel position is introduced. First, the chaotic system is used to iteratively generate a pseudo-random matrix with the same width and height as the image. When the first number of a row of the matrix is even, the elements of that row are sorted in ascending order to obtain an index vector. Conversely, when the first number of a row of the matrix is odd, the elements of that row are sorted in descending order to obtain another index vector. The corresponding rows of the image are then sorted by the corresponding index vectors. The pixel column permutation scheme follows a similar process. The implementation processes are shown in Figs. 11 and 12, respectively. After the pixel row and column permutation process, the image becomes further confused and unrecognizable.

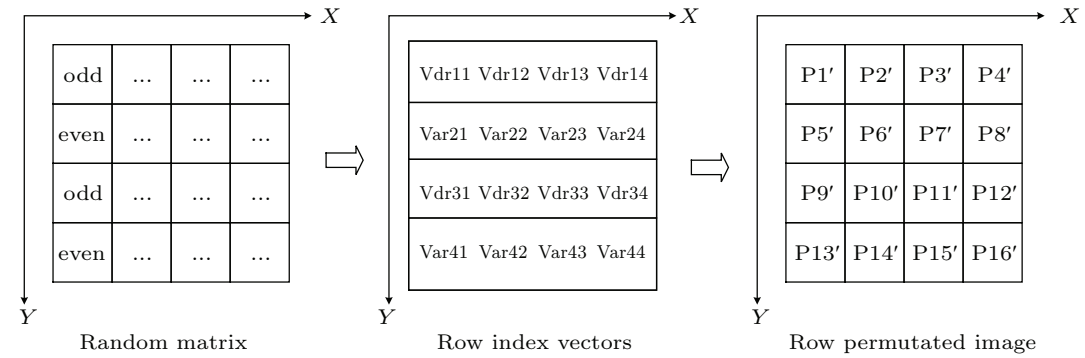


Fig. 11. The pixel position row permutation process.

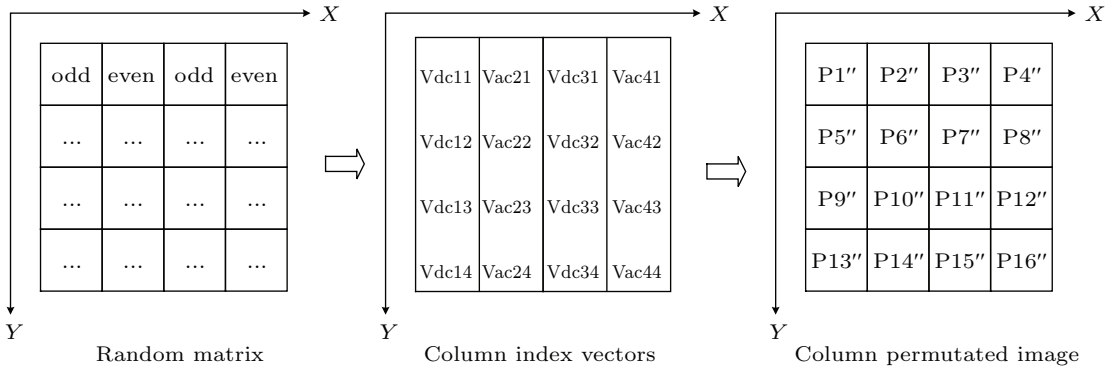


Fig. 12. The pixel position column permutation process.

6. Proposed double quantum images encryption and decryption scheme

In this section, the proposed double quantum images encryption scheme is presented in detail. The encryption process

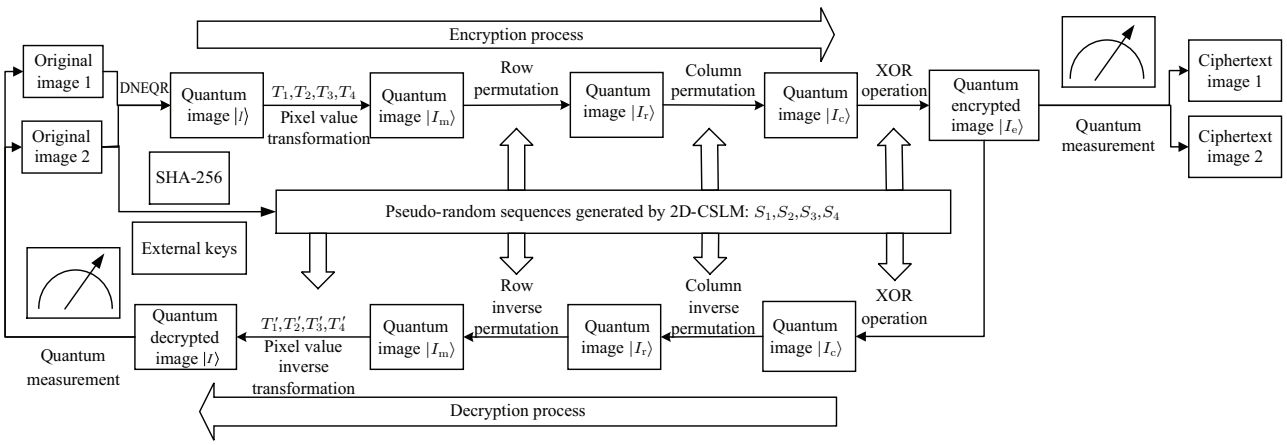


Fig. 13. Flowchart of the proposed encryption and decryption.

consists of four main steps as follows: First, the SHA-256 algorithm is applied to calculate the hash values of the two digital images and then the initial values of the chaotic system are obtained. After that, they are represented by the DNEQR model. Next, pixel value scrambling and pixel position permutation are performed on the quantum image. Finally, two XOR diffusion operations are performed to obtain the final quantum encrypted image. After quantum measurement, two ciphertext images will be obtained. The decryption process is the inverse of the encryption process. The entire flowchart is shown in Fig. 13. Specific execution steps of the proposed scheme are described below.

6.1. Double quantum images encryption scheme

Step 1 Assume that both original grayscale images P_1, P_2 have a size of $2^n \times 2^n (M \times N)$, where M and N denote the length and width of the image, respectively.

In order to increase the security, we adopt a combination of external keys and internal keys, where the external keys are one-time keys (x_{r0}, y_{r0}) sampled from the environmental noise, and the internal keys are bitwise XOR values of the hash values obtained from two plaintext images by the SHA-256 algorithm. The internal keys can be calculated by the following formula:

$$[h_i]_{i=1}^{64} = \mathrm{hex}(\mathrm{bitxor}(H_{256}(P_1), H_{256}(P_2))), \tag{12}$$

where the function $H_{256}(\cdot)$ computes the hash values of P_1 and P_2, by which two 256-bits can be obtained. The function bitoxr$(\cdot)$ performs bitwise XOR operation. The function hex$(\cdot)$ converts a binary number to its hexadecimal number. The initial values of the chaotic system are calculated according to the following equations:

$$\begin{cases} x_0 = \mathrm{mod}\left(x_{r0} + \mathrm{hex2dec}\left(\sum_{i=1}^{a} h_i\right) \times 10^{-3}, 1\right), \\ y_0 = \mathrm{mod}\left(y_{r0} + \mathrm{hex2dec}\left(\sum_{j=1}^{b} h_j\right) \times 10^{-3}, 1\right), \end{cases} \tag{13}$$

where the function hex2dec$(\cdot)$ converts a hexadecimal number to its decimal number. Function mod(A,B) calculates the remainder of dividing A by B. Parameters a and b are determined by both parties involved, and $a, b \in [32, 64]$.

Step 2 According to the DNEQR model, the two original images P_1 and P_2 are represented as quantum image $|I\rangle$ shown in Eq. (3).

Step 3 Substitute the initial values x_0, y_0 and control parameters α, μ into the 2D-CSLM system expressed in Eq. (5). Two initial sequences S_x and S_y are iteratively generated with a length of $M \times N$. To avoid transient effects, the first t numbers of each sequence are discarded and the sequences S_x and S_y are mapped to a specified range. The required pseudo-random sequences S_1, S_2, S_3, and S_4 are expressed as

$$\begin{aligned} S_1 &= \mathrm{mod}\left(\mathrm{floor}(S_x([t+1 : 2^{2n}+t]) \times 10^{15}, 4)\right) + 1, \\ S_2 &= \mathrm{mod}\left(\mathrm{floor}(S_y([t+1 : 2^{2n}+t]) \times 10^{15}, 256)\right), \\ S_3 &= \mathrm{mod}\left(\mathrm{floor}(S_x([2^{2n}+t+1 : 2^{2n+1}+t]) \times 10^{15}, N)\right) + 1, \\ S_4 &= \mathrm{mod}\left(\mathrm{floor}(S_y([2^{2n}+t+1 : 2^{2n+1}+t]) \times 10^{15}, M)\right) + 1, \end{aligned} \tag{14}$$

where the function floor$(\cdot)$ represents the operation of rounded down.

Step 4 Convert S_1 to a two-dimensional matrix $S_m = \mathrm{reshape}(S_1, 2^n, 2^n)$, where reshape$(\cdot)$ is the matrix conversion function, and the matrix S_m takes values ranging from 1 to 4. When $S_m(y,x)$ is equal to 1, 2, 3, and 4, respectively, perform T_1, T_2, T_3 and T_4 on the pixel values of the corresponding coordinates of the quantum image $|I\rangle$. The specific implementation processes are shown in Eqs. (7)–(11).

For clearer description, define a sub-operation S_T of the four transformation operations as shown below:

$$\begin{aligned} S_T = &\left(I \otimes \sum_{w=0}^{1} \sum_{y=0}^{2^n-1} \sum_{\substack{x=0 \\ wyx \neq WYX}}^{2^n-1} |w\rangle |yx\rangle \langle yx| \langle w|\right) \\ &+ T_k \otimes |W\rangle |YX\rangle \langle YX| \langle W|, \quad k = 1,2,3,4. \end{aligned} \tag{15}$$

Based on the operation S_T, the pixel-value transformation of the quantum image $|I\rangle$ can be realized as follows:

$$\begin{aligned} &S_T(|I\rangle) \\ &= S_T\left(\frac{1}{2^n} \sum_{W=0}^{1} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |P_{WYX}\rangle |W\rangle |YX\rangle\right) \\ &= \frac{1}{2^n} S_T\left(\sum_{w=0}^{1} \sum_{y=0}^{2^n-1} \sum_{\substack{x=0 \\ wyx \neq WYX}}^{2^n-1} |P_{wyx}\rangle |w\rangle |yx\rangle + |P_{WYX}\rangle |W\rangle |YX\rangle\right) \\ &= \frac{1}{2^n}\left(\sum_{w=0}^{1} \sum_{y=0}^{2^n-1} \sum_{\substack{x=0 \\ wyx \neq WYX}}^{2^n-1} |P_{wyx}\rangle |w\rangle |yx\rangle + T_k \otimes |W\rangle |YX\rangle\right). \end{aligned} \tag{16}$$

To realize the transform operations for all pixel values of the quantum image, the quantum operation U is applied to the whole quantum image $|I\rangle$ as follows:

$$\begin{aligned} U(|I\rangle) &= \prod_{W=0}^{1} \prod_{Y=0}^{2^n-1} \prod_{X=0}^{2^n-1} S_T(|I\rangle) = \prod_{W=0}^{1} \prod_{Y=0}^{2^n-1} \prod_{X=0}^{2^n-1} S_T\left(\frac{1}{2^n} \sum_{W=0}^{1} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} |P_{WYX}\rangle \otimes |W\rangle \otimes |YX\rangle\right) \\ &= \frac{1}{2^n} \sum_{W=0}^{1} \sum_{Y=0}^{2^n-1} \sum_{X=0}^{2^n-1} S_T(|P_{WYX}\rangle \otimes |W\rangle \otimes |YX\rangle) \end{aligned}$$

$$= \frac{1}{2^n}\sum_{W=0}^{1}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}\begin{cases}\left|p^{3}_{wyx}p^{\bar{6}}_{wyx}p^{1}_{wyx}p^{\bar{4}}_{wyx}p^{7}_{wyx}p^{\bar{2}}_{wyx}p^{5}_{wyx}p^{\bar{0}}_{wyx}\right\rangle|W\rangle|YX\rangle, & k=1,\\ \left|p^{\bar{7}}_{wyx}p^{2}_{wyx}p^{\bar{5}}_{wyx}p^{0}_{wyx}p^{\bar{3}}_{wyx}p^{6}_{wyx}p^{\bar{1}}_{wyx}p^{4}_{wyx}\right\rangle|W\rangle|YX\rangle, & k=2,\\ \left|p^{\bar{7}}_{wyx}p^{\bar{6}}_{wyx}p^{\bar{5}}_{wyx}p^{\bar{4}}_{wyx}p^{\bar{3}}_{wyx}p^{\bar{2}}_{wyx}p^{\bar{1}}_{wyx}p^{\bar{0}}_{wyx}\right\rangle|W\rangle|YX\rangle, & k=3,\\ \left|p^{7'}_{wyx}p^{6'}_{wyx}p^{5'}_{wyx}p^{4'}_{wyx}p^{3'}_{wyx}p^{2'}_{wyx}p^{1'}_{wyx}p^{0'}_{wyx}\right\rangle|W\rangle|YX\rangle, & k=4,\end{cases}$$

$$= \frac{1}{2^n}\sum_{W=0}^{1}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}|P'_{WYX}\rangle\otimes|W\rangle\otimes|YX\rangle = |I_{\mathrm{m}}\rangle. \tag{17}$$

$|P'_{wyx}\rangle$ denotes the pixel value after scrambling, T_1, T_2, T_3 and T_4 represent the four pixel-value transformation methods mentioned above. After pixel-value scrambling, the quantum image $|I_{\mathrm{m}}\rangle$ is obtained. The method is illustrated in Fig. 14.

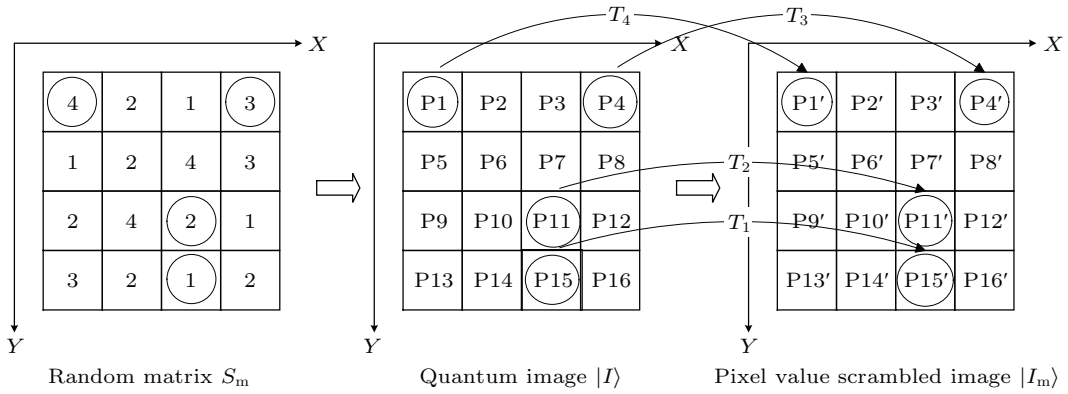


Fig. 14. The method of pixel-value scrambling.

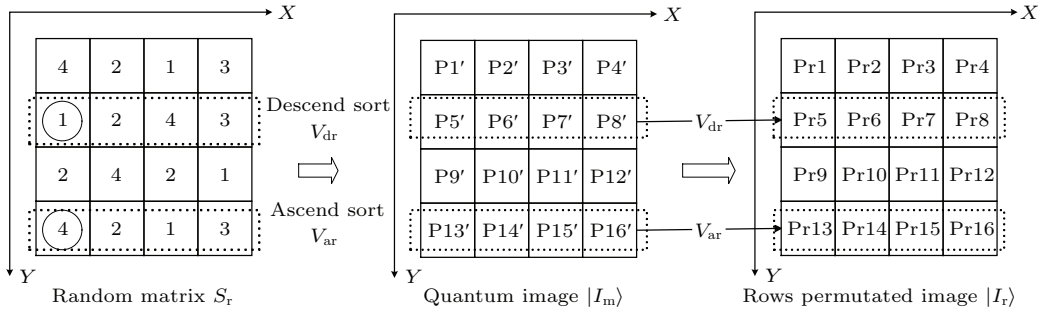


Fig. 15. The process of pixel row permutation.

Step 5 Convert S_3 into a two-dimensional matrix $S_{\mathrm{r}} = \mathrm{reshape}(S_3, 2^n, 2^n)$, and then perform pixel row permutation on the quantum image $|I_{\mathrm{m}}\rangle$ using matrix S_{r}. The specific execution process is as follows: If the first element of a row of S_{r} is even, sort the elements of that row in ascending order to obtain the index vector denoted as V_{ar}. If the first element of a row of S_{r} is odd, sort the elements of that row in descending order to obtain the index vector denoted as V_{dr}. Then arrange the corresponding rows of the image matrix $|I_{\mathrm{m}}\rangle$ according to V_{ar} or V_{dr} to obtain the row permutation image $|I_{\mathrm{r}}\rangle$. The specific realization process is shown in Fig. 15.

Step 6 Convert S_4 into a two-dimensional matrix $S_{\mathrm{c}} = \mathrm{reshape}(S_4, 2^n, 2^n)$, and then perform pixel column permutation on quantum image $|I_{\mathrm{r}}\rangle$ using matrix S_{c}. Similar to Step 5, when the first element of a column of S_{c} is even, the elements of that column are sorted in ascending order to obtain the index vector denoted as V_{ac}. When the first element of a column of S_{c} is odd, the elements of that column are sorted in descending order to obtain the index vector denoted as V_{dc}. Then arrange the corresponding columns of the image matrix $|I_{\mathrm{r}}\rangle$ according to V_{ac} or V_{dc} to obtain the column permutation image $|I_{\mathrm{c}}\rangle$. The corresponding process is shown in Fig. 16.

Step 7 The final ciphertext image can be obtained after two XOR diffusion operations. Firstly, the quantum image $|I_{\mathrm{c}}\rangle$ is taken out in alternating order and reverse order based on the row priority principle, resulting in the pixel sequence $P_{\mathrm{r}}(i), i = 0, 1, 2, \ldots, 2^{2n}-1$. The sequence $P_{\mathrm{c}}(i)$ can be obtained by performing XOR operations between $P_{\mathrm{r}}(i)$ and $S_2(i)$. Secondly, flip $S_2(i)$ over to get $S_{2\mathrm{F}}(i)$, then the sequence $P_{\mathrm{d}}(i)$ is obtained by performing XOR operations between $P_{\mathrm{c}}(i)$ and $S_{2\mathrm{F}}(i)$. The two XOR operations are illustrated by the following equations:

$$P_{\mathrm{c}}(i) = P_{\mathrm{r}}(i) \oplus S_2(i), \tag{18}$$

$$S_{2\mathrm{F}}(i) = \mathrm{fliplr}\left(S_2(i)\right), \tag{19}$$

$$P_{\mathrm{d}}(i) = P_{\mathrm{c}}(i) \oplus S_{2\mathrm{F}}(i), \tag{20}$$

where the function fliplr$(\cdot)$ denotes flipping the matrix in the left–right direction. Finally, the final encrypted image $|I_e\rangle$ is obtained by placing the sequence $P_d(i)$ into a two-dimensional matrix of size $2^n \times 2^n$ in alternating order and reverse order according to the principle of column precedence. The execution of the two-XORs operation is shown in Fig. 17.

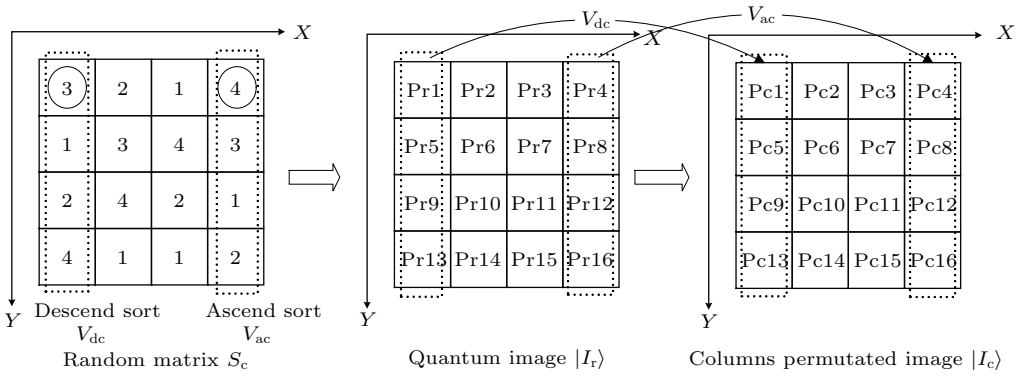


Fig. 16. The process of pixel column permutation.

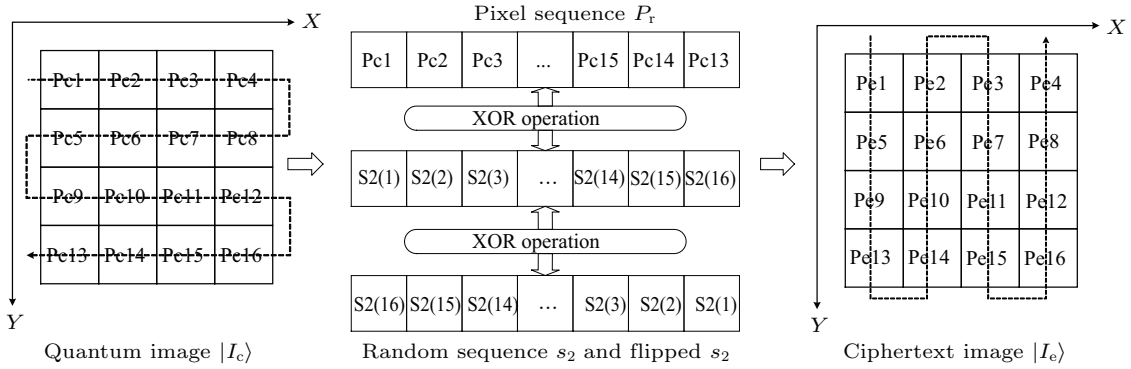


Fig. 17. The process of XOR diffusion operation.

6.2. Double quantum images decryption scheme

As quantum operations are reversible, the decryption process is the inverse of encryption. The decryption process is shown in lower part of Fig. 13.

7. Experimental simulation results and performance analysis

Since quantum computers are not available yet, a classical computer system with MATLAB 7.0 software was used to perform simulation experiments. The quantum states and quantum operations are modeled using complex vectors and unitary matrices, respectively. The external one-time keys are sampled from optimized noise: $(x_{r0}, y_{r0}) = (0.916904439913408, 0.410270206990945)$. The agreed parameter settings are $a = 32$, $b = 64$, and the initial values of 2D-CSLM are calculated according to Eqs. (12) and (13). The control parameters are set as $\alpha = 4$, $\mu = 44.99$, and the value of t is set to 1000 in Eq. (14). The encryption results are shown in Fig. 18. It can be seen that no useful information can be obtained from the encrypted images, which validates the security and effectiveness of the algorithm.

7.1. Key space analysis

The key space of the encryption algorithm should be sufficiently large to resist brute force attacks. In this paper, the control parameter α is fixed, while the other three parameters x_0, y_0, and μ, are the keys. The precision of the initial values and control parameters for 2D-CSLM is set to 10^{-15}. As a result, the total key space amounts to $(10^{15})^3 = 10^{45} \approx 2^{149} > 2^{100}$. Therefore, the encryption scheme has a sufficiently large key space to resist brute force attacks.

7.2. Key sensitivity analysis

Key sensitivity means that a small change in the key has a significant impact on the encrypted or decrypted image. During the decryption process, using an incorrect key, even with a slight difference from the correct key, fails to recover the original image. In this study, four different key combinations are employed to decrypt the image as illustrated in Table 2. The image "peppers" is selected as an example to demonstrate the key sensitivity. The decrypted image using the correct key, denoted as K, is shown in Fig. 19(a). The decrypted images using the wrong keys K_1, K_2, and K_3 are shown in Figs. 19(b)–19(d). It is evident that even with a slight alteration in the key, the decrypted image is unrecognizable, indicating the high sensitivity and its resistance against brute force attacks.

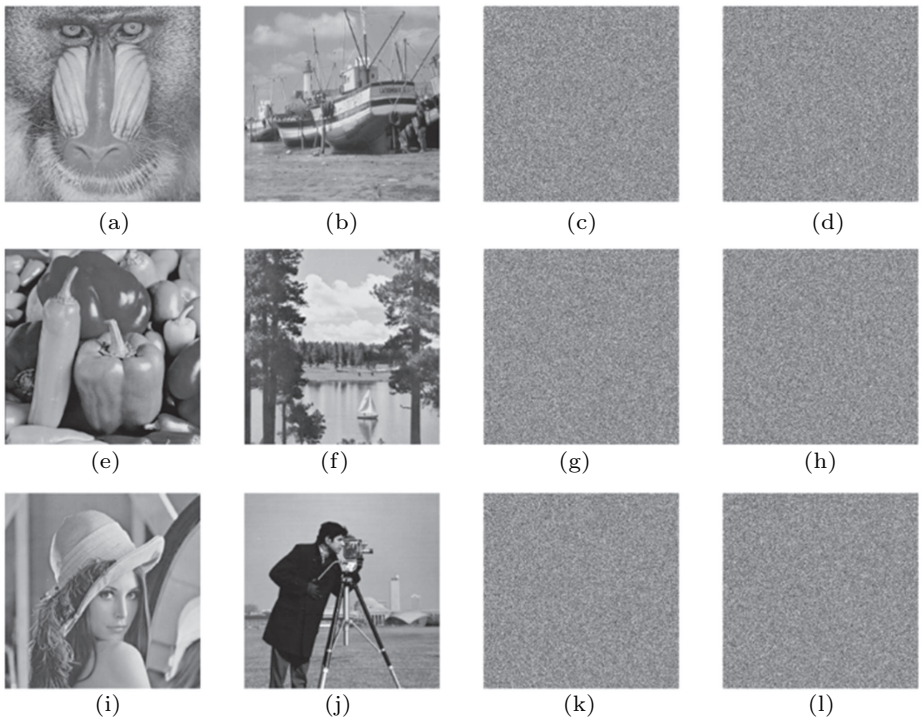


Fig. 18. Three pairs of original grayscale images and corresponding encrypted images: (a) baboon, (b) boat, (c) encrypted baboon, (d) encrypted boat, (e) peppers, (f) lake, (g) encrypted peppers, (h) encrypted lake, (i) Lena, (j) cameraman, (k) encrypted Lena, (l) encrypted cameraman.

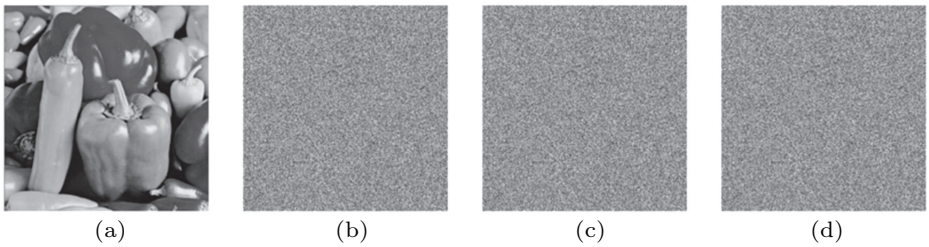


Fig. 19. (a) Decryption with correct key K, (b) decryption with incorrect key K_1, (c) decryption with incorrect key K_2, (d) decryption with incorrect key K_3.

Table 2. Four different key combinations.

K	K_1	K_2	K_3
x_0, y_0, μ	x_0+10^{-15}, y_0, μ	x_0, y_0+10^{-15}, μ	$x_0, y_0, \mu+10^{-15}$

7.3. Histogram analysis

The histogram provides statistical information about the image, illustrating the distribution of each gray value in the

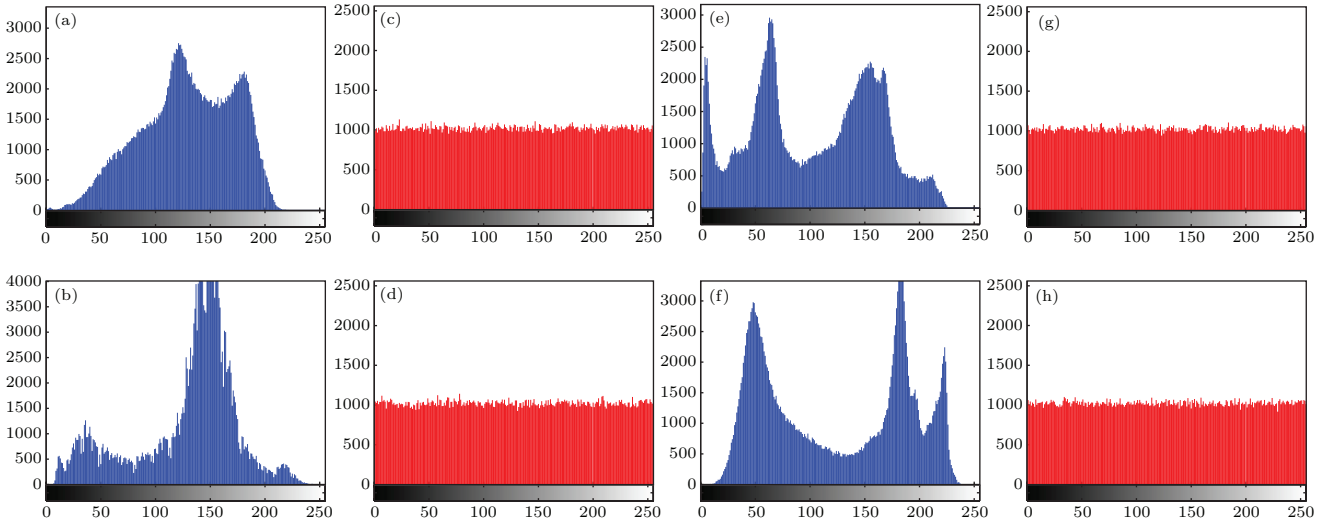


Fig. 20. Histograms of (a) baboon, (b) boat, (c) encrypted baboon, (d) encrypted boat, (e) peppers, (f) lake, (g) encrypted peppers, (h) encrypted lake.

image. Figures 20(a), 20(b), 20(e), and 20(f) display the histograms of original images, while Figs. 20(c), 20(d), 20(g), and 20(h) show the corresponding histograms of ciphertext images. It is evident that there are distinct differences between the histograms before and after encryption. The histograms of the original images exhibit an uneven distribution, while the histograms of the ciphertext images are uniformly distributed and similar to each other. Therefore, the proposed encryption algorithm exhibits enhanced resistance against statistical analysis attacks.

7.4. Correlation of adjacent pixels

Generally speaking, the correlation between adjacent pixels in plaintext images is high, and a reliable encryption algorithm should make the correlation between adjacent pixels in ciphertext images as close to zero as possible. To analyze the correlation, 10000 pairs of adjacent pixels are randomly selected from the original and encrypted images of the "boat" in horizontal, vertical, and diagonal directions. The correlation distribution in three directions is shown in Fig. 21. It can be seen that the correlation of the encrypted image is significantly reduced and evenly distributed in all directions. The correlation coefficient (CC) of adjacent pixels can be calculated using the following formula:[47]

$$\mathrm{CC} = \frac{\sum_{i=1}^{L}\left(x_i - \frac{1}{L}\sum_{i=1}^{L} x_i\right)\left(y_i - \frac{1}{L}\sum_{i=1}^{L} y_i\right)}{\sqrt{\sum_{i=1}^{L}\left(x_i - \frac{1}{L}\sum_{i=1}^{L} x_i\right)^2 \sum_{i=1}^{L}\left(y_i - \frac{1}{L}\sum_{i=1}^{L} y_i\right)^2}}, \tag{21}$$

where x_i, y_i are the values of adjacent pixels, and L is the total number of selected adjacent pixels.

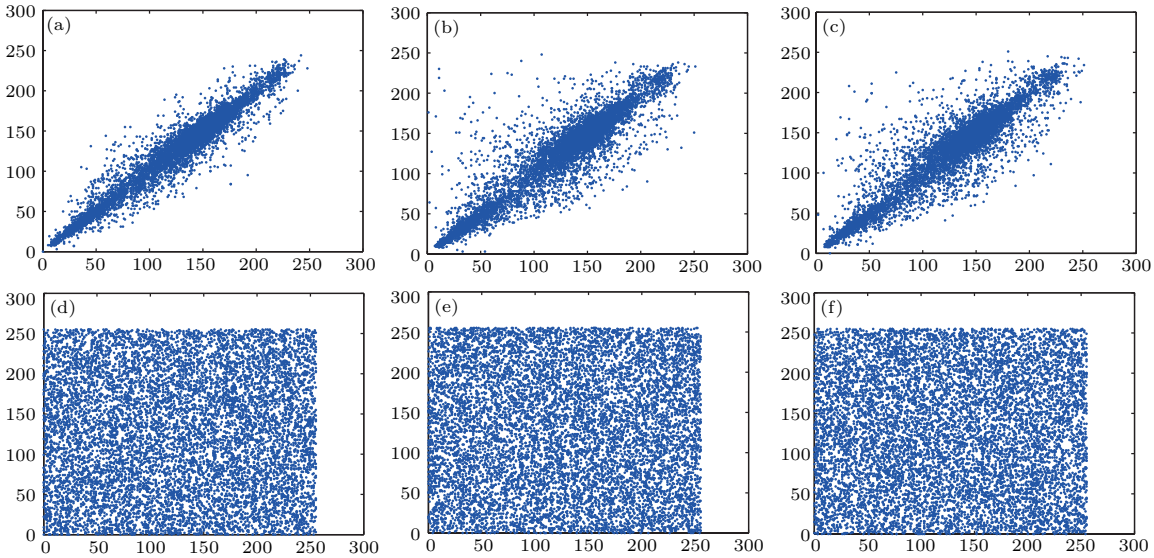


Fig. 21. Correlation distribution in three directions: (a)–(c) original image of "boat", (d)–(f) encrypted image of "boat".

Table 3. CC values of original and encrypted images in three directions.

Images	Image size: 256×256			Image size: 512×512		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Baboon	0.7541	0.6191	0.6010	0.8642	0.7603	0.7353
Encrypted baboon	**0.0019**	**–0.0074**	**0.0017**	**0.0036**	**–0.0055**	**0.0008**
Boat	0.8783	0.9173	0.8295	0.9449	0.9725	0.9301
Encrypted boat	**0.0010**	**–0.0002**	**0.0005**	**–0.0006**	**0.0021**	**–0.0043**
Peppers	0.9554	0.9623	0.9310	0.9804	0.9820	0.9676
Encrypted peppers	**–0.0025**	**0.0023**	**–0.0033**	**–0.0060**	**0.0058**	**–0.0016**
Lake	0.9393	0.9389	0.9080	0.9775	0.9750	0.9627
Encrypted lake	**0.0033**	**0.0025**	**0.0016**	**0.0065**	**0.0016**	**–0.0009**
Lena	0.9246	0.9586	0.9063	0.9714	0.9844	0.9607
Encrypted Lena	**–0.0008**	**0.0060**	**–0.0055**	**0.0058**	**0.0015**	**–0.0010**
Cameraman	0.9571	0.9618	0.9265	0.9802	0.9905	0.9718
Encrypted cameraman	**–0.0020**	**–0.0036**	**–0.0013**	**–0.0030**	**0.0045**	**–0.0048**
House	0.8869	0.8926	0.8174	0.9507	0.9642	0.9218
Encrypted house	**0.0077**	**–0.0031**	**–0.0087**	**–0.0026**	**0.0085**	**–0.0027**
Barbara	0.8081	0.8784	0.8366	0.8639	0.9613	0.8434
Encrypted Barbara	**0.0068**	**0.0001**	**–0.0073**	**0.0023**	**0.0063**	**0.0043**
Ref. [40]	–	–	–	0.0008	–0.0074	–0.0060
Ref. [48]	0.0295	0.0187	0.0393	–	–	–
Ref. [49]	–0.0041	–0.0126	–0.0228	–	–	–

The CC values are presented in Table 3, with the CC values of the encrypted images highlighted in bold. It can be observed that the CC values in all three directions are close to zero for the encrypted images. This indicates that the proposed encryption algorithm can provide sufficient security against correlation analysis attacks.

7.5. Information entropy

The information entropy (IE) measures the degree of diffusion for each gray level in an image. It serves as a significant indicator to assess the capability of an image encryption system in resisting attacks. For a grayscale image, the optimal value of information entropy is 8 bits, which is calculated using the following formula:

$$\mathrm{IE}(x) = -\sum_{i=0}^{H} P(x_i)\log_2 P(x_i), \tag{22}$$

where H denotes the total number of pixel levels in the image, x_i is the gray value, and $P(x_i)$ is the probability of the gray value x_i. As demonstrated in Table 4, the information entropy of the encrypted image is close to 8, indicating its effectiveness in resisting entropy attacks.

Table 4. Information entropy of the original images and encrypted images.

Images	Image size: 256×256		Image size: 512×512	
	Original	Encrypted	Original	Encrypted
Baboon	7.3658	7.9972	7.3587	7.9993
Boat	7.1944	7.9970	7.1952	7.9991
Peppers	7.5706	7.9971	7.5727	7.9993
Lake	7.4789	7.9973	7.4839	7.9994
Lena	7.4448	7.9971	7.4464	7.9993
House	7.2365	7.9972	7.2374	7.9993
Ref. [12]	7.1894	7.9966	–	–
Ref. [50]	7.2374	7.9239	–	–
Ref. [51]	–	–	7.4845	7.9987

7.6. Differential attack

A robust image encryption algorithm should exhibit sensitivity to the plaintext image, such that even a small change in pixel values can yield two distinct ciphertext images. Differential attacks are a form of selective plaintext, and the resistance to such attacks relies on the algorithm's sensitivity to the plaintext. Two metrics, namely the number of pixel change rate (NPCR) and the uniform average change intensity (UACI), quantify this, the sensitivity. To ensure effectiveness, the NPCR and UACI should be maximized,[52] as defined below:

$$\mathrm{NPCR} = \frac{\sum_{i,j} D(i,j)}{M\times N}\times 100\%,$$
$$D(i,j) = \begin{cases} 0, & C_1(i,j) = C_2(i,j), \\ 1, & C_1(i,j) \neq C_2(i,j), \end{cases} \tag{23}$$
$$\mathrm{UACI} = \frac{1}{M\times N}\left[\sum_{i,j}\frac{|C_1(i,j)-C_2(i,j)|}{255}\right]\times 100\%, \tag{24}$$

where M and N represent the width and height of the two ciphertext images. $C_1(i,j)$, and $C_2(i,j)$ correspond to the ciphertext images obtained before and after changing a pixel value of the plaintext image, respectively.

By randomly changing pixel values of the grayscale images, the corresponding NPCR and UACI values can be obtained, as shown in Table 5. It can be seen that the value of NPCR exceeds 99% and the value of UACI surpasses 33%. This indicates that even minor changes to the original image result in significantly different ciphertext images. Thus, the proposed encryption algorithm exhibits superior resistance to differential attacks.

Table 5. Results of NPCR and UACI.

Images	Image size	NPCR (%)	UACI (%)
Baboon	256×256	99.6094	33.7260
Boat	256×256	99.6384	33.5857
Peppers	256×256	99.6323	33.4789
Lake	512×512	99.6128	33.4617
Lena	512×512	99.6223	33.5265
Cameraman	512×512	99.6075	33.5052
Ref. [7]	512×512	99.6273	32.4228
Ref. [53]	512×512	99.6187	33.4969
Ref. [54]	256×256	99.6102	33.3915

7.7. Cutting attack

In the process of image transmission, it is possible that a part of the data may be cropped or lost. So, the encryption algorithm should possess a robust ability to counteract such cropping effectively and deal with the damaged ciphertext image. As depicted in Figs. 22(a)–22(d), where 1/16, 1/8, 1/4, and 1/2 parts of the ciphertext image "boat" are cropped and then decrypted, it is clear that most of the information still can be recovered, as shown in Figs. 22(e)–22(h). Hence, the encryption scheme can effectively withstand cutting attacks.

7.8. Noise attack

Images are usually subjected to noise interference during storage and transmission, which can impact the decryption process of the image. Therefore, secure encryption algorithms must be robust enough against noise attacks. As shown

in Figs. 23(a)–23(d), the ciphertext image of "house" is decrypted by adding 0.05, 0.1, 0.2, and 0.3 salt and pepper noise, respectively. The results show that most of the information still can be recovered, as depicted in Figs. 23(e)–23(h). So, the encryption scheme is able to effectively resist the noise attack and has excellent robustness.

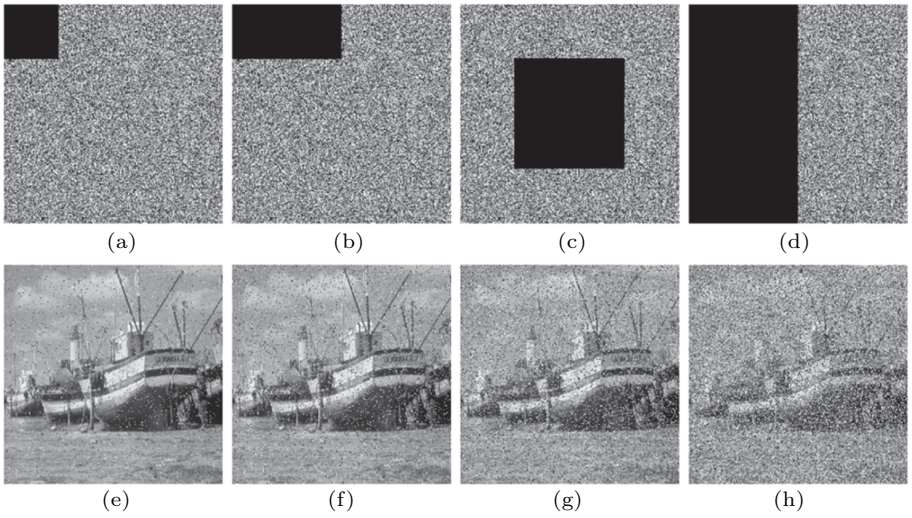


Fig. 22. Results of cutting attack on the ciphertext image "boat": (a) cut off 1/16, (b) cut of 1/8, (c) cut off 1/4, (d) cut off 1/2; (e)–(h) the decrypted images of (a)–(d), respectively.

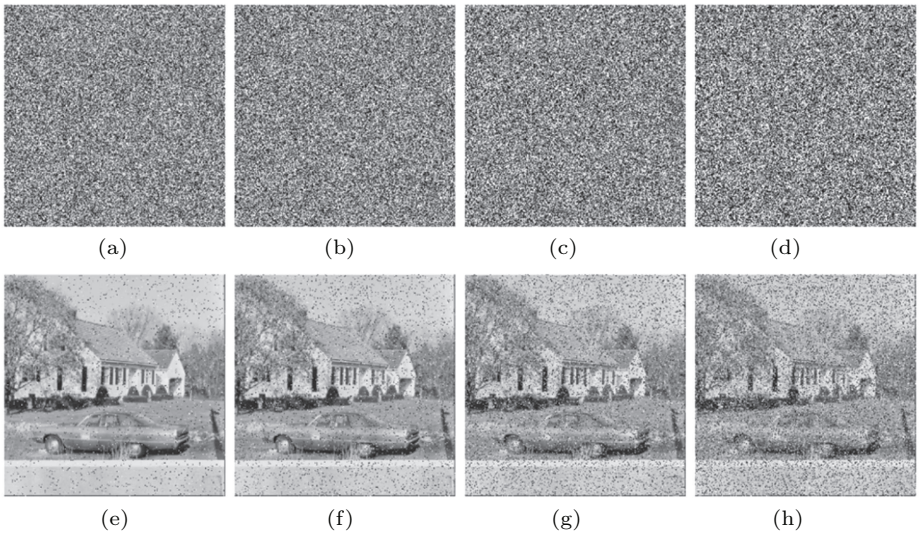


Fig. 23. Results of noise attack on the ciphertext image "house": (a)–(d) adding 0.05, 0.1, 0.2, and 0.3 of salt and pepper noise, respectively; (e)–(h) the decrypted images of (a)–(d), respectively.

8. Conclusion

In this paper, a novel double quantum images representation model, DNEQR, is proposed, which greatly reduces the number of qubits and improves the efficiency of encryption. A two-dimensional hyperchaotic system is constructed based on sine and logistic maps, resulting in a large parameter space and improved chaotic properties. Additionally, based on the DNEQR model and the proposed two-dimensional hyperchaotic system, a method for encryption and decryption of double quantum images is designed. During encryption, four pixel-value scrambling methods are proposed to randomly scramble the pixel values of the image. The rows and columns of pixels are separately permutated to enhance the confusion effect of the image. Finally, a pseudo-random sequence is utilized to carry out two XOR diffusion operations on the permutated quantum image, significantly increasing the confusion of pixel values. Simulation experiments and performance analysis demonstrate that the proposed encryption algorithm exhibits superior security and effectiveness. Further research will focus on refining and enhancing the current algorithm by exploring the representation model and encryption scheme for double quantum images.

Data availability statement

The data that support the findings of this study are openly available in Science Data Bank at https://doi.org/10.57760/sciencedb.j00113.00185.

Acknowledgements

Project supported by the Open Fund of Anhui Key Laboratory of Mine Intelligent Equipment and Technology (Grant No. ZKSYS202204), the Talent Introduction Fund of Anhui University of Science and Technology (Grant No. 2021yjrc34), and the Scientific Research Fund of Anhui Provincial Education Department (Grant No. KJ2020A0301).

References

[1] Cai J R, Gu S H and Zhang L 2018 *IEEE Trans. Image Process.* **27** 2049
[2] Li Y C, Zhou R G, Xu R Q, Luo J and Hu W W 2020 *Quantum Sci. Technol.* **5** 044003
[3] Guo L M, Du H W and Huang D 2021 *Quantum Inf. Process.* **21** 20
[4] Lai Q, Liu Y and Yang L 2023 *Appl. Intell.* **53** 22863
[5] Sharkawy N H, Afify Y M, Gad W and Badr N 2022 *IEEE Access* **10** 63004
[6] Yao S Y, Chen L F and Zhong Y 2019 *Opt. Laser Technol.* **120** 105703
[7] Deepak Vagish K, Rajakumaran C and Kavitha R 2020 *Multimedia Tools Appl.* **79** 23849
[8] Shi J J, Chen S H, Chen T, Zhao T G, Tang J Q, Li Q, Yu C L and Shi H Y 2022 *Quantum Inf. Process.* **21** 214
[9] Su Y N and Wang X Y 2023 *Multimedia Tools Appl.* **82** 42679
[10] Wang J, Geng Y C, Han L and Liu J Q 2019 *Int. J. Theor. Phys.* **58** 308
[11] Zhou S H 2020 *IEEE Access* **8** 178336
[12] Liu X B, Xiao D and Liu C 2020 *Quantum Inf. Process.* **19** 239
[13] Le P Q, Iliyasu A M, Dong F and Hirota K 2011 *Quantum Inf. Process.* **10** 63
[14] Zhang Y, Lu K, Gao Y H and Wang M 2013 *Quantum Inf. Process.* **12** 2833
[15] Lisnichenko M and Protasov S 2022 *Quantum Mach. Intell.* **5** 2
[16] Li H S, Zhu Q X, Zhou R G, Song L and Yang X J 2014 *Quantum Inf. Process.* **13** 991
[17] Jiang N, Wang J and Yue M 2015 *Quantum Inf. Process.* **14** 4001
[18] Li H S, Song S X, Fan P, Peng H L and Liang Y 2019 *Inf. Sci.* **502** 42
[19] Li H S, Chen X, Xia H Y, Liang Y and Zhou Z S 2018 *IEEE Access* **6** 62396
[20] Abd-El-Atty B, Abd-El-Latif A A and Venegas-Andraca S E 2019 *Quantum Inf. Process.* **18** 272
[21] Chen Z G, Yan Y, Pan J and Zhu H H 2022 *Quantum Inf. Process.* **21** 175
[22] Li H S, Li C Y, Chen X and Xia H Y 2018 *Int. J. Theor. Phys.* **57** 3745
[23] Malik A, Dhall S and Gupta S 2021 *Multimedia Tools Appl.* **80** 7911
[24] Zhu H H, Chen X B and Yang Y X 2021 *Quantum Inf. Process.* **20** 315
[25] Zhou R G, Wu Q, Zhang M Q and Shen C Y 2013 *Int. J. Theor. Phys.* **52** 1802
[26] Hou C G, Liu X B and Feng S Y 2020 *Mod. Phys. Lett. A* **35** 2050145
[27] Gao J, Wang Y N, Song Z Y and Wang S M 2023 *Entropy* **25** 865
[28] Fan P, Hou M J, Hu W W and Xiao K 2022 *Int. J. Theor. Phys.* **61** 260
[29] Yang Y G, Xia J, Jia X and Zhang H 2013 *Quantum Inf. Process.* **12** 3477
[30] Gong L H, He X T, Tan R C and Zhou Z H 2018 *Int. J. Theor. Phys.* **57** 59
[31] Li H S, Li C Y, Chen X and Xia H Y 2019 *Mod. Phys. Lett. A* **34** 1950214
[32] Wang L, Ran Q W and Ma J 2020 *Multimedia Tools Appl.* **79** 6661
[33] Liu X B, Xiao D and Liu C 2018 *Entropy* **20** 867
[34] Jiang N, Dong X, Hu H, Ji Z X and Zhang W Y 2019 *Int. J. Theor. Phys.* **58** 979
[35] Zhou R G and Li Y B 2020 *Int. J. Quantum Inf.* **18** 2050022
[36] Wang Y, Chen L, Yu K, Gao Y and Ma Y 2022 *Entropy* **24** 251
[37] Jiang Z Y and Liu X B 2023 *Int. J. Theor. Phys.* **62** 22
[38] Chen X, Yu S M, Wang Q X, Guyeux C and Wang M J 2023 *Multimedia Tools Appl.* **82** 42717
[39] Zhang J and Huo D 2019 *Multimedia Tools Appl.* **78** 15605
[40] Hu M T, Li J Q and Di X Q 2023 *Nonlinear Dyn.* **111** 2815
[41] Khairullah M K, Alkahtani A A, Bin Baharuddin M Z and Al-Jubari A M 2021 *Electronics* **10** 2116
[42] Hua Z Y, Zhou Y C, Pun C M and Chen C L P 2015 *Inf. Sci.* **297** 80
[43] Zhu H, Zhao Y and Song Y 2019 *IEEE Access* **7** 14081
[44] Hua Z Y and Zhou Y C 2016 *Inf. Sci.* **339** 237
[45] Tang J N, Zhang F and Ni H 2023 *The Visual Computer* **39** 4955
[46] Pareschi F, Rovatti R and Setti G 2012 *IEEE Trans. Inf. Forensics Secur.* **7** 491
[47] Faragallah O S, Afifi A, El-Shafai W, El-Sayed H S, Naeem E A, Alzain M A, Al-Amri J F, Soh B and El-Samie F E A 2020 *IEEE Access* **8** 42491
[48] Liu X B, Xiao D and Liu C 2021 *Quantum Inf. Process.* **20** 23
[49] Liu X B, Xiao D and Xiang Y P 2019 *IEEE Access* **7** 6937
[50] Chen Z, Yan Y, Pan J and Zhu H 2022 *Quantum Inf. Process.* **21** 175
[51] Li X Z, Chen W W and Wang Y Q 2018 *Int. J. Theor. Phys.* **57** 2904
[52] Kaur M and Kumar V 2020 *Arch. Comput. Methods Eng.* **27** 15
[53] Patro K A K, Kumar M P and Acharya B 2022 *Sādhanā* **47** 161
[54] Liu H, Zhao B and Huang L Q 2019 *Multimedia Tools Appl.* **78** 20465