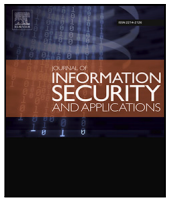
Journal of Information Security and Applications 81 (2024) 103698

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa



Quantum image encryption algorithm via optimized quantum circuit and parity bit-plane permutation

Jinwen He, Hegui Zhu *, Xv Zhou

College of Sciences, Northeastern University, Shenyang, 110819, China

ARTICLE INFO

Keywords:
Quantum image encryption
BRQI quantum representation
Optimized quantum circuit
Bit-level

ABSTRACT

Quantum image encryption technology employs the unique features of superposition, entanglement, and quantum state instability, offering advantages like high efficiency, parallelism, and robust resistance to decryption attempts. In this work, we propose a quantum image encryption algorithm via an optimized quantum circuit and parity bit-plane permutation named OCPBP. Our research commences with applying an optimization algorithm to BRQI image preparation. This optimization significantly reduces the number of auxiliary qubits from n+3 to 2 in the preparation of (n+5)-CNOT gate preparation. This reduction makes a significant simplification of the complexity and memory space required for BRQI image preparation. Moreover, we introduce an innovative operation based on parity pixel bit-planes for permutation, which provides a fresh perspective on quantum image encryption based on the BRQI model. Furthermore, we generate quantum key images for XOR operations using the improved logistic map, effectively tackling the challenge of non-random keys that can be a concern in conventional BRQI encryption techniques. We also perform a row-column-based permutation operation to enhance the algorithm's resilience against potential attacks. Through extensive theoretical analysis and comparative experiments, we confirm the heightened security and reduced complexity of the proposed OCPBP algorithm compared to BRQI-based and chaotic image encryption methods. It holds significant promise for improving quantum image communication and application.

1. Introduction

Image is an integral part of multimedia data in network communication, and image security has been paid more attention. Various encryption technologies have appeared, some stream ciphers such as DES, AES, and RSA are proposed for text encryption. However, unlike traditional text information, images possess distinct characteristics, including substantial data capacity and strong pixel correlation, which render traditional encryption algorithms like DES, AES, and RSA unsuitable for image encryption again because of the encryption efficiency [1–3]. With the development of image technology and methods, many excellent image encryption algorithms have been proposed [4–8].

In the realm of digital image encryption methods reliant on computer-based encryption, limited by traditional processes and the continuous expansion of the electronic device integration scale, quantum effects will inevitably affect the performance of component devices in integrated circuits. In this case, Moore's law will eventually fail [9], which limits the subsequent development of classical calculations [10]. The current possible solution to the problem of Moore's law failure is to use different computational modes, and scientists are focusing on quantum computing. With the continuous development of quantum science and the improvement of quantum theory, the encryption technology based on quantum information theory has gradually drawn academic attention. Due to the entanglement and superposition characteristics of the quantum state, quantum encryption technology has the advantages of parallelism, difficulty to crack, robust security, and high encryption speed. This has gradually made quantum encryption an essential choice for image encryption. Some outstanding quantum algorithms with excellent encryption performance [11–13] have been put forward. However, there are only a few studies in quantum image encryption since the common quantum image representation techniques cannot store bit-plane information individually.

Quantum image processing primarily delves into exploring quantum image algorithms tailored for quantum computers, offering valuable insights into image processing techniques. Research in quantum image processing can be broadly categorized into two main branches. The first category entails the application of concepts and theories from the realm of quantum mechanics to process classical images. This approach primarily focuses on classical images, employing quantum mechanics techniques to improve existing image processing algorithms or devise new ones. It is important to emphasize that this category does

* Corresponding author.
E-mail address: zhuhegui@mail.neu.edu.cn (H. Zhu).

https://doi.org/10.1016/j.jisa.2024.103698

Available online 13 January 2024

Fig. 1. General Process of Quantum Image Processing. The blue border represents the digital matrix image, while the purple border represents the quantum state image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Comparison of other Quantum image representations for a gray-scale image of 2^{2n} pixels.

QIR	Qubits	Image size	Pixel encoding
FRQI [14]	2n+1	$2^{2n-k} \times 2^k$	Amplitude
NEQR [15]	2n+8	$2^n \times 2^n$	Basis states
INEQR [16]	2n+8	$2^{2n-k} \times 2^k$	Basis states
GQIR [17]	2n+8	$2^{2n-k} \times 2^k$	Basis states
BRQI [18]	2n+4	$2^{2n-k} \times 2^k$	Basis states

not strictly involve quantum algorithms but encompasses quantum-derived image processing algorithms. The second category of quantum image processing algorithms revolves around transforming a classical image into a quantum format, which can be manipulated using specific matrix transformations. Typically, the converted image is stored in classical computers as a quantum state represented as a column vector. Subsequently, a series of operations are performed on this quantum state through specific unitary transformations. After obtaining the final quantum state, quantum measurements are conducted, converting the image back to a classical matrix format for output. A depiction of the quantum image encryption algorithm process can be found in Fig. 1 . The algorithm proposed in this article aligns with the second type of quantum image processing and has undergone simulation experiments on classical computers to mimic the quantum encryption process.

The premise of the quantum encryption algorithm is to perform a quantum representation of images and then perform a series of encryption operations on quantum images. Therefore, in 2011, the flexible representation of quantum images (FRQI) [14] was proposed to map the color and position of an image to a quantum state. Later, a new enhanced quantum image representation (NEQR) [15] used the qubit sequence's ground state to store each pixel's gray-scale value. Unlike FRQI, a sequence of two entangled quantum bits stores an integer image representing all pixel's pixel and position values in NEQR. However, the problem with NEQR is that there is a strict requirement for selecting the original image size, which only applies to images of square format. Based on this, Nan et al. [16] made appropriate adjustments to the NEQR model and developed the improved NEQR named INEQR. Moreover, the generalized quantum image representation (GQIR) [17] was proposed based on improved NEQR, which can represent color images. In 2018, Li et al. [18] proposed a bit-plane-based quantum image representation (BRQI), which can use qubits to store the image position, bit-plane, and color channel information. To illustrate the advantages of BRQI, Table 1 compares five quantum image representations (QIRs) to BRQI. The results show that although FRQI requires fewer qubits to represent the image, its pixel encoding is probabilistic amplitude, and the storage capacity is much smaller than that of the basis states encoding model. In contrast to other QIRs using basis states coding, BRQI can not only represent rectangular images but also has minimal bit consumption.

As various quantum image representations were proposed, image encryption algorithms based on different quantum image models emerged endlessly. In 2012, Akhshani et al. [19] proposed an image encryption approach based on the logistic map, which became the first to use chaos map combined with quantum image encryption. Later, Yang et al. [20,21] proposed a new quantum image encryption approach based on quantum Fourier transform and double random phase encoding theory. In 2018, Zhang et al. [22] combined quantum theory and DNA encryption to implement image diffusion dynamically. In recent years, Zhou et al. [23] proposed a novel quantum image compression and encryption algorithm based on DQWT and a 3D hyper-chaotic Henon map in 2020. In 2021, Ye et al. [24] proposed an asymmetric image encryption approach based on the quantum logistic map and cyclic modulo diffusion. In quantum image processing, due to the high complexity of chaotic systems and their initial solid value sensitivity, chaos theory is often combined with quantum image encryption algorithms to ensure the efficiency of encryption.

Since BRQI represents the bit-planes with three qubits, it makes it possible for the bit-plane encryption operation to enter the field of quantum encryption. Mahsa et al. [25] proposed a color image encryption method based on the BRQI model combined with the encryption method proposed in Ref. [18], which used a specific image as the key for an encryption operation. Shahrokh et al. [26] explored a quantum selective encryption for medical images combining BRQI with image selection. Although the BRQI model introduces bit-plane encryption into quantum image encryption, a problem has been realized. The problem is the non-random keys in existing methods. Both Ref. [25] and Ref. [26] use specific circuits or images for encryption operations, and neither uses random keys for encryption. In addition, because BRQI divides the image information at the bit-plane level, more auxiliary quantum bits are required to prepare quantum images, which significantly increases memory occupation and time consumption. Moreover, the encryption algorithm based on the BRQI model only performs bit-plane-level encryption operations, and the encryption process is relatively simple.

In general, the existing BRQI-based approaches have the following drawbacks: (1) the existing algorithms do not use random keys for encryption operations; (2) the preparation of BRQI quantum images consumes memory and has high time complexity; (3) the existing algorithms only perform encryption operations at the bit-plane level, and the encryption methods are relatively simple.

Aiming at solving these problems, this study proposes a quantum image encryption algorithm via an optimized quantum circuit and parity bit-plane permutation named OCPBP, which realizes quantum image encryption at the bit-plane level through the BRQI model. Then, it employs a chaotic map to generate the key image, thus solving the non-random problem of the key. First, OCPBP addresses the issue of the high complexity of quantum image preparation by offering a preparation optimization circuit for BRQI images. Moreover, we provide a new parity pixel bit-plane permutation algorithm that modifies the pixel values by changing the bit-plane. To solve the issue that the key cannot be randomized, we construct quantum key images using chaotic signals [19] and then perform XOR operations on the bit-plane permuted images. Finally, we perform a row-column-based permutation operation to complete the second permutation operation. The main technical contributions are summarized below:

- Proposing an optimization method for the BRQI image preparation circuit, which achieves a remarkable reduction in the number of auxiliary qubits from n+3 to just 2 for the preparation of (n+5)-CNOT gates, and minimizes the complexity and memory space of BRQI preparation.
- Proposing a parity pixel bit-plane-based permutation operation combined with pixel-level encryption, which extends the BRQI-based encryption approach in the bit-plane encryption field.

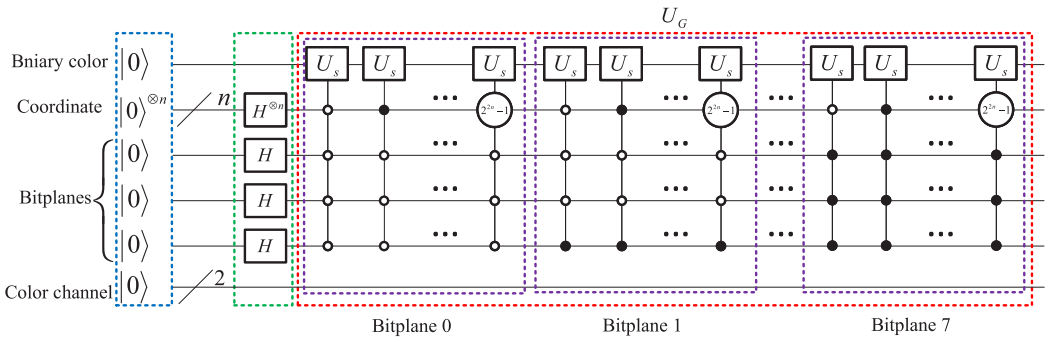


Fig. 2. Single-channel BRQI image preparation circuit. Complete the process of $|\Psi_1\rangle \rightarrow |\Psi^8\rangle$.

- Exploring the quantum key images for XOR operations with chaotic map, effectively tackling the challenge of non-random keys that can be a concern in conventional BRQI encryption techniques.

The remainder of this paper is structured as follows. In Section 2, we introduce the knowledge of the BRQI model. In Section 3, we illustrate the detailed content of the proposed OCPBP, which consists of an optimization approach for quantum image preparation, a new bit-plane-based permutation algorithm, a quantum key image-based-diffusion operation, and a row-column-based permutation operation. The simulation performance and the complexity analysis of the OCPBP are illustrated in Section 4. Section 5 verifies the performance of OCPBP regarding the efficacy and security of various other algorithms. Section 6 summarizes the advantages and characteristics of OCPBP compared with other similar algorithms. Finally, we summarize the main findings in Section 7.

2. BRQI image preparation

Li et al. [18] first introduced the bit-plane representation quantum images model (BRQI), which can store the gray-scale and RGB images with the size of $2^{n-k} \times 2^k$, $k(0 < k < n)$. Since the proposed encryption approach is mainly based on RGB image encryption, we briefly describe the process of BRQI to store color images in the following part.

Generally, the RGB image can be divided into three channel images for storage. The image in each color channel can be regarded as a gray-scale image, and each gray-scale image can be decomposed into 8-bit-plane images. BRQI divides the position and color information of the image into bit-plane images for storage, and each bit-plane can be considered separately as a binary image. A single channel image with the size of $2^{n-k} \times 2^k$ can be expressed as

$$\left|\Psi^8\right\rangle = \frac{1}{\sqrt{2^{n+3}}} \sum_{l=0}^{2^3-1} \sum_{x=0}^{2^{n-k}-1} \sum_{y=0}^{2^k-1} |g(x,y)\rangle|x\rangle|y\rangle|l\rangle, \tag{1}$$

then, an RGB image with the size of $2^{n-k} \times 2^k$ can be expressed as

$$\left|\Psi^{24}\right\rangle = \frac{1}{\sqrt{2^{n+3}}} \sum_{ch=1}^{3} \sum_{l=0}^{2^3-1} \sum_{x=0}^{2^{n-k}-1} \sum_{y=0}^{2^k-1} |g(x,y)\rangle|x\rangle|y\rangle|l\rangle|ch\rangle, \tag{2}$$

where $|x\rangle = |x_0, x_1 \ldots x_{n-k-1}\rangle$ and $|y\rangle = |y_0, y_1 \ldots y_{k-1}\rangle$ store the horizontal and vertical coordinate information of the input image, respectively. $|l\rangle$ is the lth bit-plane, $g(x,y) \in \{0,1\}$ is the binary color of the bit-plane image, $|ch\rangle$ indicates the RGB channel and can take the values of 01, 10 and 11.

According to Eq. (2), for a $2^{n-k} \times 2^k$ RGB image, BRQI requires n+6 qubits for preparation, where n qubits represent the image size, one qubit represents the color information, three qubits represent the bit-plane information and two qubits represent the color channel information.

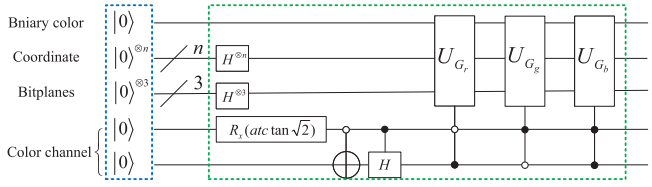


Fig. 3. Quantum circuit of RGB image. The blue dotted frame completes the initialization process of the quantum register $|\Psi_0\rangle$, the green dotted frame completes the process of $|\Psi_1\rangle \rightarrow |\Psi^{24}\rangle$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Eq. (3) depicts the preparation procedure of RGB image, which realizes the preparation process from the all-zero vector $|\Psi_0\rangle$ to the quantum state $\left|\Psi^{24}\right\rangle$.

$$\begin{aligned} &|\Psi_0\rangle \xrightarrow{U_1} |\Psi_1\rangle = \frac{1}{\sqrt{2^{n+3}}} \sum_{l=0}^{2^3-1} \sum_{x=0}^{2^{n-k}-1} \sum_{y=0}^{2^k-1} |0\rangle|x\rangle|y\rangle|l\rangle|0\rangle^{\otimes 2} \rightarrow \\ &\left|\Psi^8\right\rangle \rightarrow \left|\Psi^{24}\right\rangle = \frac{1}{\sqrt{3}} \left(\left|\Psi^R\right\rangle|01\rangle + \left|\Psi^G\right\rangle|10\rangle + \left|\Psi^B\right\rangle|11\rangle\right) \end{aligned} \tag{3}$$

First, a quantum register with the size of n+6 is used to initialize the quantum state $|\Psi_0\rangle$ to complete the storage process of $2^{n-k} \times 2^k$ image. Let the quantum state $|\Psi_0\rangle = |0\rangle^{\otimes n+6}$, where $|0\rangle^{\otimes n+6}$ represents the tensor product of order n+6 of $|0\rangle$.

Next, we give the circuit realization of color image storage combined with the quantum circuit. Fig. 2 shows the quantum circuit diagram of a single channel, where the green dotted frame indicates the use of the operator $U_1 = I \otimes H^{\otimes n+3} \otimes I^{\otimes 2}$ acting on $|\Psi_0\rangle$ to obtain a superposition state $|\Psi_1\rangle$. This process can be realized by

$$\begin{aligned} |\Psi_1\rangle &= U_1|\Psi_0\rangle = I \otimes H^{\otimes n+3} \otimes I^{\otimes 2} \left(|0\rangle^{\otimes n+6}\right) \\ &= \frac{1}{\sqrt{2^{n+3}}} \sum_{l=0}^{2^3-1} \sum_{x=0}^{2^{n-k}-1} \sum_{y=0}^{2^k-1} |0\rangle|x\rangle|y\rangle|l\rangle|0\rangle^{\otimes 2}, \end{aligned} \tag{4}$$

where matrix I is a second-order identity matrix, H is the Hadamard gate, and $H^{\otimes k}$ represents the tensor product of order k of H.

The U_G module in the red dotted frame in Fig. 2 completes the process of $|\Psi_1\rangle \rightarrow \left|\Psi^8\right\rangle$. The role of U_s is to realize the color information assignment, and can be defined by

$$U_s = (g(x,y) \oplus 1)I + g(x,y)X, \tag{5}$$

where X is NOT gate, I is identity operator. If the pixel value is 0, then $U_s = I$. If the pixel value is 1, then $U_s = X$.

Since the RGB image can be stored in R, G, and B channels, the quantum circuit of the color image can be obtained by the operation shown in Fig. 3 after completing the three single-channel image storage.

The blue dotted frame in Fig. 3 completes the initialization process of quantum register $|\Psi_0\rangle$, and the green dotted frame completes the process of $|\Psi_1\rangle \rightarrow \left|\Psi^{24}\right\rangle$, where U_{G_r}, U_{G_g} and U_{G_b} represent the single

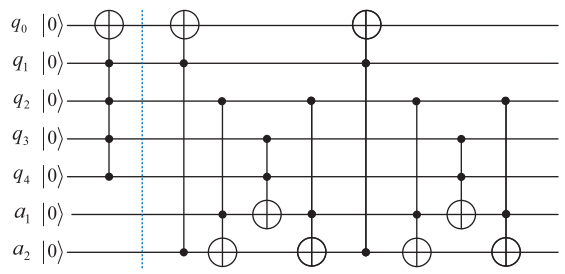


Fig. 4. Example of 4-CNOT circuit decomposition. The preparation decomposition circuit of this 4-CNOT gate is shown on the right side of the dashed line.

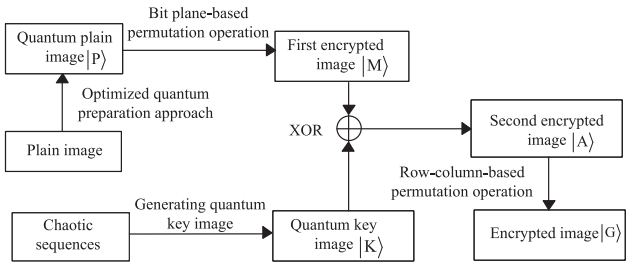


Fig. 5. The structure of the proposed OCPBP, which consists of twice permutation operations and once diffusion operation.

channel process shown in Fig. 3. The whole circuit completes the preparation process from the all-zero vector $|\Psi_0\rangle$ to the quantum state $|\Psi^{24}\rangle$.

According to Eq. (4), for a $2^{n-k} \times 2^k$ image, we need n bits to represent the position $|xy\rangle$, 3 bits to represent the binary color bit-plane $|l\rangle$, 2 bits to represent the color channel $|ch\rangle$ and 1 bit to represent the binary color $|c\rangle$. Therefore, we need a total of n+5 bits to represent the position information of the image and 1 bit to represent the binary color. Therefore, a total of n+6 qubits are required to represent RGB images, and n+4 qubits are required to represent gray-scale images.

As depicted in Figs. 2 and 3, it is evident that the m-CNOT gate is frequently utilized in the preparation process. Taking a 4-CNOT gate as an example, Fig. 4 illustrates its decomposition into a quantum circuit comprising basic quantum logic gates, where $q_0, \ldots, q_4$ are the control circuit, a_1 and a_2 are auxiliary qubits. It can be seen that the preparation process of a 4-CNOT gate can be decomposed into 8 Toffli gates, and two auxiliary bits are required.

According to Ref. [27], each m-CNOT gate can be decomposed into 4m-8 Toffoli gates with m-2 auxiliary qubits, where m is an arbitrary integer. In this way, for a $2^{n-k}\times 2^k$ image, since the location information needs (n+5)-CNOT gate to be stored by BRQI, the circuit preparation process requires (n+5)-2=n+3 auxiliary qubits.

3. The proposed OCPBP

The structure of the proposed OCPBP is depicted in Fig. 5, which consists of twice permutation operations and once diffusion operation. First, we use the optimized quantum image preparation method to prepare the plain image into BRQI quantum image $|P\rangle$ through the optimized quantum image preparation method and obtain the first encrypted image $|M\rangle$ through the bit-plane-based permutation operation. Then, we conduct the XOR operation with the quantum key image generated by the chaotic sequence and the encrypted image $|M\rangle$ and obtain the diffused image $|A\rangle$. Finally, we use a new row-column-based permutation operation to permute the image $|A\rangle$ for the second time to obtain the final encrypted image $|G\rangle$.

3.1. Optimization of BRQI quantum circuits

As discussed in Section 2, for a $2^{n-k} \times 2^k$ RGB image, the circuit preparation process of the (n+5)-CNOT gates requires n+3 auxiliary qubits. Since the auxiliary qubits occupy much memory space, we employ the reset operation with the idea described in Ref. [28] to recover the auxiliary qubits, which can reduce the auxiliary qubits from n+3 to 2. Now we show the optimization process for the preparation of all (n+5)-CNOT gates in the quantum image preparation circuit shown in Figs. 2 and 3. a_1 and a_2 are auxiliary qubits, which will be used in the preparation of (n+5)-CNOT gates. The initial value is a_1=0, a_2=1, and the optimized preparation of BRQI quantum circuits is illustrated in the following steps.

Step 1: For the initial quantum state $|\Psi_0\rangle = |0\rangle^{\otimes n+6}$, we employ the operator U_1 on it to obtain the superposition state $|\Psi_1\rangle$ by Eq. (4).

Step 2: Use operator U_X to change all the qubits values of 0 in the location information to 1 by

$$U_X|q_i\rangle = ((|q_i\rangle \oplus |1\rangle)X + |q_i\rangle I)|q_i\rangle \; i = 1, \ldots, n+5, \tag{6}$$

where X is NOT gate, I is the identity operator, $\oplus$ means XOR operation and $|q_i\rangle$ ($i = 1, 2, \ldots, n+5$) are the n+5 qubits containing position, bit-plane, and color channel information.

Step 3: Use Toffoli gate for n+5 times to transfer the state of $|q_i\rangle$ to auxiliary qubits. The location information can be transferred through the following operations

$$\begin{cases} |a_1\rangle = |a_1\rangle \oplus (|a_2\rangle|q_i\rangle) & i \; is \; an \; odd \; number, \\ |a_2\rangle = |a_2\rangle \oplus (|a_1\rangle|q_i\rangle) & i \; is \; an \; even \; number. \end{cases} \tag{7}$$

Note that in each transmission, the auxiliary bit in the uncontrolled bit of the Toffoli gate is reset to 0. After finishing the n+5 transfers, all bit states representing location information have been transferred, and the value of the currently controlled auxiliary bit is 1.

Step 4: If the initial values of location bit q_i is 0, we use NOT gate to restore it by $|q_i\rangle = X(|q_i\rangle)$ $i = 1, \ldots, n+5$.

Step 5: In the plain image, if the binary color q_0 of the bit-plane image at the current position is 1, then we take the control bit of the last Toffoli gate in Step 3 as the control bit and use U_F to change q_0 to 1 by

$$U_F|q_0\rangle = ((|a_k\rangle \oplus |1\rangle)I + |a_k\rangle X)|q_0\rangle \; k = (n+5) mod \; 2. \tag{8}$$

If $a_k = 1$, then $U_F|q_0\rangle = X|q_0\rangle = |1\rangle$. Otherwise, the U_F operation is not used. Finally, reset a_k=0 for preparation at the next location.

The reset operation in Step 3 resets the auxiliary qubit to 0, making the auxiliary qubit reusable. This allows for an optimized method to prepare an m-CNOT gate using only two auxiliary bits ($|a_1\rangle$, $|a_2\rangle$). In contrast, the unoptimized preparation process for a (n+5)-CNOT gate, as stored by BRQI, requires n+3 auxiliary qubits. However, with the proposed optimization, only **two auxiliary qubits** are needed for the preparation process, which can significantly reduce the demand for qubits and storage space while gradually completing the state transfer process of the quantum image.

Moreover, after applying the proposed optimized circuit to the quantum image preparation process described in Section 2, we can describe the digital image as a quantum image $|P\rangle$ by

$$|P\rangle = \frac{1}{\sqrt{2^{n+3}}} \sum_{ch=1}^{3} \sum_{l=0}^{2^3-1} \sum_{x=0}^{2^{n-k}-1} \sum_{y=0}^{2^k-1} |c\rangle|x\rangle|y\rangle|l\rangle|ch\rangle, \tag{9}$$

where $|x\rangle$ and $|y\rangle$ store the position information of the input image, respectively. $|l\rangle$ is the lth bit-plane, $|c\rangle$ is the binary color of the bit-plane image, $|ch\rangle$ indicates the RGB channel.

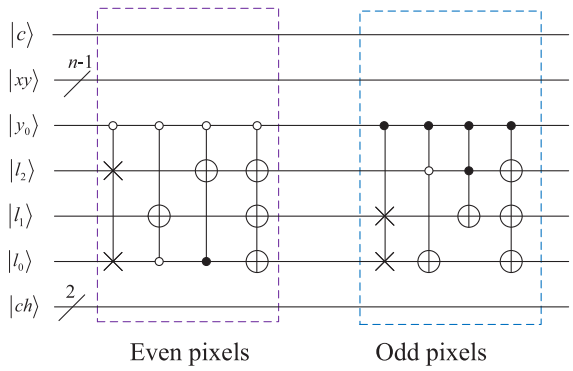


Fig. 6. Quantum circuit of bit-plane-based permutation. By altering the state of the 3 qubits representing the bit-plane information, the color information of the pixel is modified. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Even pixels	Odd pixels
$\|000\rangle \to \|000\rangle \to \|010\rangle \to \|101\rangle$	$\|000\rangle \to \|000\rangle \to \|001\rangle \to \|110\rangle$
$\|001\rangle \to \|100\rangle \to \|110\rangle \to \|001\rangle$	$\|001\rangle \to \|010\rangle \to \|011\rangle \to \|100\rangle$
$\|010\rangle \to \|010\rangle \to \|000\rangle \to \|111\rangle$	$\|010\rangle \to \|001\rangle \to \|000\rangle \to \|111\rangle$
$\|011\rangle \to \|110\rangle \to \|100\rangle \to \|011\rangle$	$\|011\rangle \to \|011\rangle \to \|010\rangle \to \|101\rangle$
$\|100\rangle \to \|001\rangle \to \|101\rangle \to \|010\rangle$	$\|100\rangle \to \|100\rangle \to \|110\rangle \to \|001\rangle$
$\|101\rangle \to \|101\rangle \to \|001\rangle \to \|110\rangle$	$\|101\rangle \to \|110\rangle \to \|100\rangle \to \|011\rangle$
$\|110\rangle \to \|011\rangle \to \|111\rangle \to \|000\rangle$	$\|110\rangle \to \|101\rangle \to \|111\rangle \to \|000\rangle$
$\|111\rangle \to \|111\rangle \to \|011\rangle \to \|100\rangle$	$\|111\rangle \to \|111\rangle \to \|101\rangle \to \|010\rangle$

Fig. 7. Permutation process of bit-plane. By applying the quantum circuit from Fig. 6 for encryption, the bit-plane information is altered.

3.2. Parity pixel bit-plane-based permutation operation

After preparing the quantum image $|P\rangle$, we perform its first permutation operation. The main goal of this operation is to change the pixel's color information by changing three qubits that represent the bit-plane information. Fig. 6 illustrates the detailed quantum circuit of the permutation operation, where $|xy\rangle$ is the pixel position, $|y_0\rangle$ is the basis for dividing odd and even pixels, $|l_i\rangle$ is the bit-plane, $|ch\rangle$ is the color channel, and $|c\rangle$ is the binary color.

For even pixels ($|y_0\rangle$=0), we first employ the SWAP gate to exchange the information of $|l_0\rangle$ and $|l_2\rangle$. Next, according to the value of $|l_0\rangle$, it can be divided into the following two cases. If $|l_0\rangle$=0, which means the bit-planes are 0, 2, 4, 6, then do NOT operation on $|l_1\rangle$. Otherwise, if $|l_0\rangle$=1, do NOT operation on $|l_2\rangle$. Finally, do NOT operation on $|l_0\rangle$, $|l_1\rangle$ and $|l_2\rangle$, respectively.

For odd pixels ($|y_0\rangle$=1), we also use the SWAP gate to exchange the information of $|l_0\rangle$ and $|l_1\rangle$. Next, with the value of $|l_2\rangle$, The operation can be divided into the following two cases. If $|l_2\rangle$=0, which means the bit-planes are 0, 1, 2, 3, then do NOT operation on $|l_0\rangle$. Otherwise, if $|l_2\rangle$=1, do NOT operation on $|l_1\rangle$. Finally, do NOT operation on $|l_0\rangle$, $|l_1\rangle$ and $|l_2\rangle$, respectively.

To illustrate the details of this operation, Fig. 7 depicts the bit-plane transformation procedure of the algorithm, where the second, third, and fourth columns of the process correspond to the first, second, and third operations in the preceding text, respectively.

Take the $|100\rangle$ as an example. For even pixels, $|l_0\rangle$ and $|l_2\rangle$ are exchanged through the SWAP gate. Since $|l_0\rangle$ is 1 at this time, then $|l_2\rangle$ is set to 1 through the NOT operation. Then do NOT operation on $|l_0\rangle$, $|l_1\rangle$ and $|l_2\rangle$ to get $|010\rangle$. For odd pixels, $|l_0\rangle$ and $|l_1\rangle$ are exchanged through the SWAP gate. Since $|l_2\rangle$ is 1 at this time, l_1 is set to 1 by NOT operation. Finally, do NOT operation on $|l_0\rangle$, $|l_1\rangle$, $|l_2\rangle$ and get $|001\rangle$.

Fig. 8 shows the effect of the proposed permutation algorithm. It can be seen that the bit-plane-based permutation operation can not only disrupt the position of binary pixels but also change the color of pixels.

Now we apply the above bit-plane-based permutation operation to the quantum image $|P\rangle$ obtained in Section 3.1 and then get the first encrypted image $|M\rangle$, which can be expressed by

$$|P\rangle \xrightarrow{\text{permute } |l\rangle} |M\rangle = \frac{1}{\sqrt{2^{n+3}}} \sum_{ch=1}^{3} \sum_{l=0}^{2^3-1} \sum_{x=0}^{2^{n-k}-1} \sum_{y=0}^{2^k-1} |c\rangle|x\rangle|y\rangle|\bar{l}\rangle|ch\rangle, \tag{10}$$

where $|\bar{l}\rangle$ represents bit-plane information after permutation.

3.3. Quantum key image-based-diffusion operation

3.3.1. Generate key images using improved logistic map

Using chaotic signals in the encryption process can effectively increase the key's randomness and the encrypted image's security level. Therefore, many chaotic system [29,30] and excellent chaotic encryption algorithms [31–33] have been proposed. Though the bit-plane-based permutation operation has changed the pixel value, it cannot randomize the key. Therefore, a quantum key image-based-diffusion operation is employed to solve this problem.

In the diffusion operation, we used an improved logistic map [19], which is expressed as:

$$\begin{cases} x = \mu x(1-x) \\ y = \arcsin\left(\text{sqrt}\left(x\right)\right)/\pi, \end{cases} \tag{11}$$

where $\mu \in [3.681, 4]$.

To evaluate the chaotic behavior of this mapping, we plotted the bifurcation diagram and the Lyapunov exponent image of the improved logistic map in Fig. 9. It can be seen that when $\mu \in [3.681, 4]$, the improved Logistic map appears chaotic.

For a $2^{n-k}\times 2^k$ image, let $M = 2^{n-k}$, $N = 2^k$, and five key parameters μ_1, μ_2, $x_1(1)$, $x_2(1)$, λ are utilized to construct the key stream sequence on each of the RGB channels, the detailed procedure to construct key image is shown as follows.

Step 1 Use the improved logistic map to generate two sequences y_1, y_2 with the length of $M \times N$ by

$$\begin{cases} x_k(i) = \mu_k x_k(i-1)\left(1 - x_k(i-1)\right) \\ y_k(i) = \arcsin\left(\text{sqrt}\left(x_k(i)\right)\right)/\pi, \end{cases} \tag{12}$$

where $k \in \{0, 1\}$, $\mu_k \in [3.681, 4]$ and $i = 2, 3, \ldots, M \times N$.

Step 2 The random chaotic sequence K distributed from 0 to 255 is obtained by modulus calculation with the following operation

$$K(m) = \begin{cases} \text{round}\left(y_1(m) \times \lambda\right) \bmod 256, & m \textit{ is an odd number}, \\ \text{round}\left(y_2(m) \times \lambda\right) \bmod 256, & m \textit{ is an even number}, \end{cases} \tag{13}$$

where $m \in [1, M\times N]$ and λ is the key parameter, which must be greater than 512.

Step 3 According to Step 1 and Step 2, we generate three keystream sequences. Then, get the gray image matrix of three color channels by

$$\begin{cases} K_R(m) \to K_R(x, y), \\ K_G(m) \to K_G(x, y), \\ K_B(m) \to K_B(x, y), \end{cases} \tag{14}$$

where $x = \lceil m/N \rceil + 1$, $y = m \bmod N$.

Then, combine the image matrix of the three channels into an RGB image and store it as a BRQI image, and the preparation process can be expressed as

$$\left.\begin{array}{l} K_R(x, y) = K(x, y, 1) \\ K_G(x, y) = K(x, y, 2) \\ K_B(x, y) = K(x, y, 3) \end{array}\right\} K_{RGB} \to |K\rangle, \tag{15}$$

$$|K\rangle = \frac{1}{\sqrt{2^{n+3}}} \sum_{ch=1}^{3} \sum_{l=0}^{2^3-1} \sum_{x=0}^{2^{n-k}-1} \sum_{y=0}^{2^k-1} |s\rangle|x'\rangle|y'\rangle|l'\rangle|ch'\rangle, \tag{16}$$

where $|s\rangle$ is the binary color of the key image.

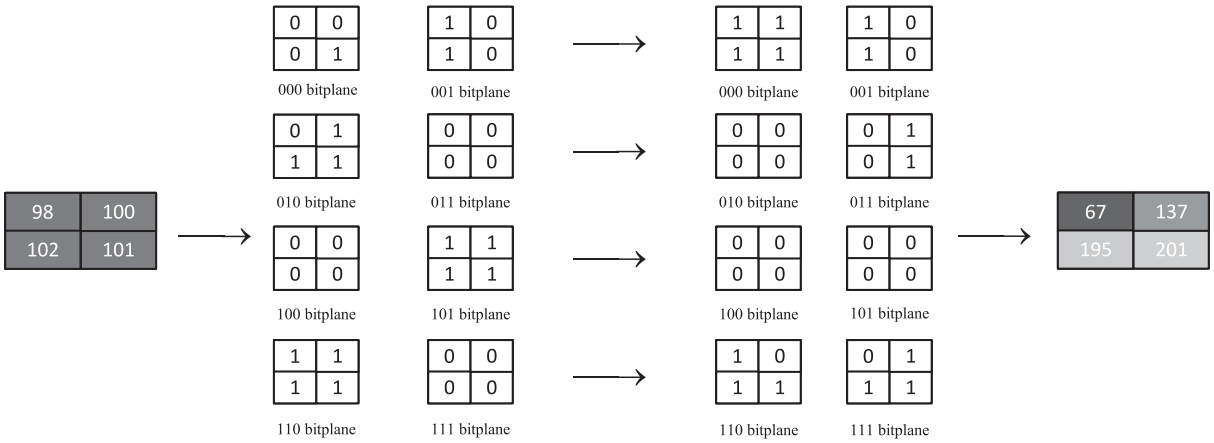


Fig. 8. Example of bit-plane-based permutation operations. After decomposing the gray-scale image into bit-planes, perform a bit-plane permutation operation, and then reassemble the encrypted bit-planes into gray-scale images.

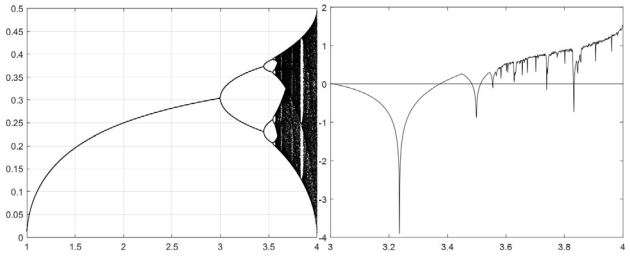


Fig. 9. Chaotic behavior: (a) Bifurcation map; (b) Lyapunov exponent spectrum analysis.

3.3.2. Diffusion operation using quantum key image

After the key image $|K\rangle$ is prepared into the BRQI image, we use the CNOT gate to complete the diffusion effect through the XOR operation with the key image $|K\rangle$ and the permuted image $|M\rangle$. The XOR operation is denoted as

$$\begin{aligned}|M\rangle \oplus |K\rangle &= \frac{1}{\sqrt{2^{n+3}}} \sum_{ch=1}^{3} \sum_{l=0}^{2^3-1} \sum_{x=0}^{2^{n-k}-1} \sum_{y=0}^{2^k-1} |c\rangle|x\rangle|y\rangle|\bar{l}\rangle|ch\rangle \\ &\oplus \frac{1}{\sqrt{2^{n+3}}} \sum_{ch=1}^{3} \sum_{l=0}^{2^3-1} \sum_{x=0}^{2^{n-k}-1} \sum_{y=0}^{2^k-1} |s\rangle|x'\rangle|y'\rangle|l'\rangle|ch'\rangle \\ &= \frac{1}{\sqrt{2^{n+3}}} \sum_{ch=1}^{3} \sum_{l=0}^{2^3-1} \sum_{x=0}^{2^{n-k}-1} \sum_{y=0}^{2^k-1} |c \oplus s\rangle|x''\rangle|y''\rangle|l''\rangle|ch''\rangle = |A\rangle.\end{aligned} \tag{17}$$

Now we give the quantum circuit in Fig. 10 to realize the XOR operation of quantum image $|M\rangle$ and key image $|K\rangle$, which is made up of the following three blocks.

In block 1, we use n+5 CNOT gates for the alignment operation to the location of quantum permuted image $|M\rangle$ and the key image $|K\rangle$. If $|XY\rangle_M = |XY\rangle_K$, in accordance with the features of CNOT gates, the location of the key image will be changed to $|0\rangle^{\otimes n+5}$, where $|XY\rangle$ represent the n+5 qubits containing position, bit-plane, and color channel information.

In block 2, the auxiliary quantum bit $|\alpha\rangle$ is combined with an initial state of $|0\rangle$. If $|\alpha\rangle$ is $|1\rangle$, then $|XY\rangle_K$ are in the state $|0\rangle^{\otimes n+5}$, which means $|XY\rangle_M = |XY\rangle_K$.

In block 3, if $|\alpha\rangle$ is $|1\rangle$, perform the XOR operation on the color information $|c\rangle$ of the permuted image $|M\rangle$ and the color information $|s\rangle$ of the key image $|K\rangle$ to obtain the quantum encrypted image $|A\rangle$.

3.4. Row-column based permutation operation

After twice-bit level encryption, we utilize a pixel-level algorithm to disrupt the location of pixels. In the encryption process below, A_c and K_c are the encrypted image and key image in the same color channel in Eq. (17), and its size is $M \times N$. For each color channel of the RGB image, perform the following encryption operation and obtain the final RGB encrypted image G.

The permutation of rows: First, we connect $A_c(x, y)$ after each $K_c(x, y)$, where $x = 1, 2, \ldots, M$, $y = 1, 2, \ldots, N$. The resulting image is denoted as T_c.

Second, for each bit of the binary color represented by $T_c(x, y)$, the XOR operation is performed by

$$T_c(x, y, l) = T_c(x, y, l) \oplus T_c(x, y, l-1) \quad l = 2, \ldots, 16, \tag{18}$$

where l represents the binary color of the l-th bit-plane.

Third, sort $T_c(x, y)$ of each row from small to large, and obtain the resulting image T_{c1} by taking the last 8 bits of $T_c(x, y)$ by

$$T_{c1}(x, y, l) = T_c(x, y, l+4) \quad l = 1, \ldots, 4. \tag{19}$$

The permutation of columns: First, connect $T_{c1}(x, y)$ after each $K_c(x, y)$, and get the resulting image by T_{c2}.

Second, for each bit of the binary color represented by $T_{c2}(x, y)$, the XOR operation is performed by

$$T_{c2}(x, y, l) = T_{c2}(x, y, l) \oplus T_{c2}(x, y, l-1) \quad l = 2, \ldots, 16. \tag{20}$$

Third, sort $T_{c2}(x, y)$ of each column from small to large, and obtain the final encrypted image in single color channel G_c by taking the last 8 bits of $T_{c2}(x, y)$ by

$$G_c(x, y, l) = T_{c2}(x, y, l+4) \quad l = 1, 2, 3, 4. \tag{21}$$

To give a clearer explanation, Fig. 11 provides an example of a single color channel image. K_c and A_c are the key image and encrypted image of the same color channel. Taking $A_c(2, 1)$ as an example, we first connect $A_c(2, 1)$ after $K_c(2, 1)$ and get $T_c(2, 1)$, then perform the XOR operation for each binary color bit to obtain 1111101011110111, with a decimal value of 64247. Next, sort T_c of each row to get the sorted position $T_c(2, 4)$ and the binary color of $T_{c1}(2, 4)$ is obtained by taking the last eight digits. After that, perform the column transformation. First, connect $T_{c1}(2, 4)$ after $K_c(2, 4)$. Then, perform the XOR operation for each binary color to get 1001000101011010, with a decimal value of 37210. Finally, sort T_{c1} of each column to get the sorted position $T_{c1}(3, 2)$ and the binary color of $G_c(3, 2)$ is obtained after permutation by taking the last eight digits.

As shown in Fig. 11, the row-column-based permutation operation can change the color value of pixels while displacing pixel positions and has a good encryption effect.

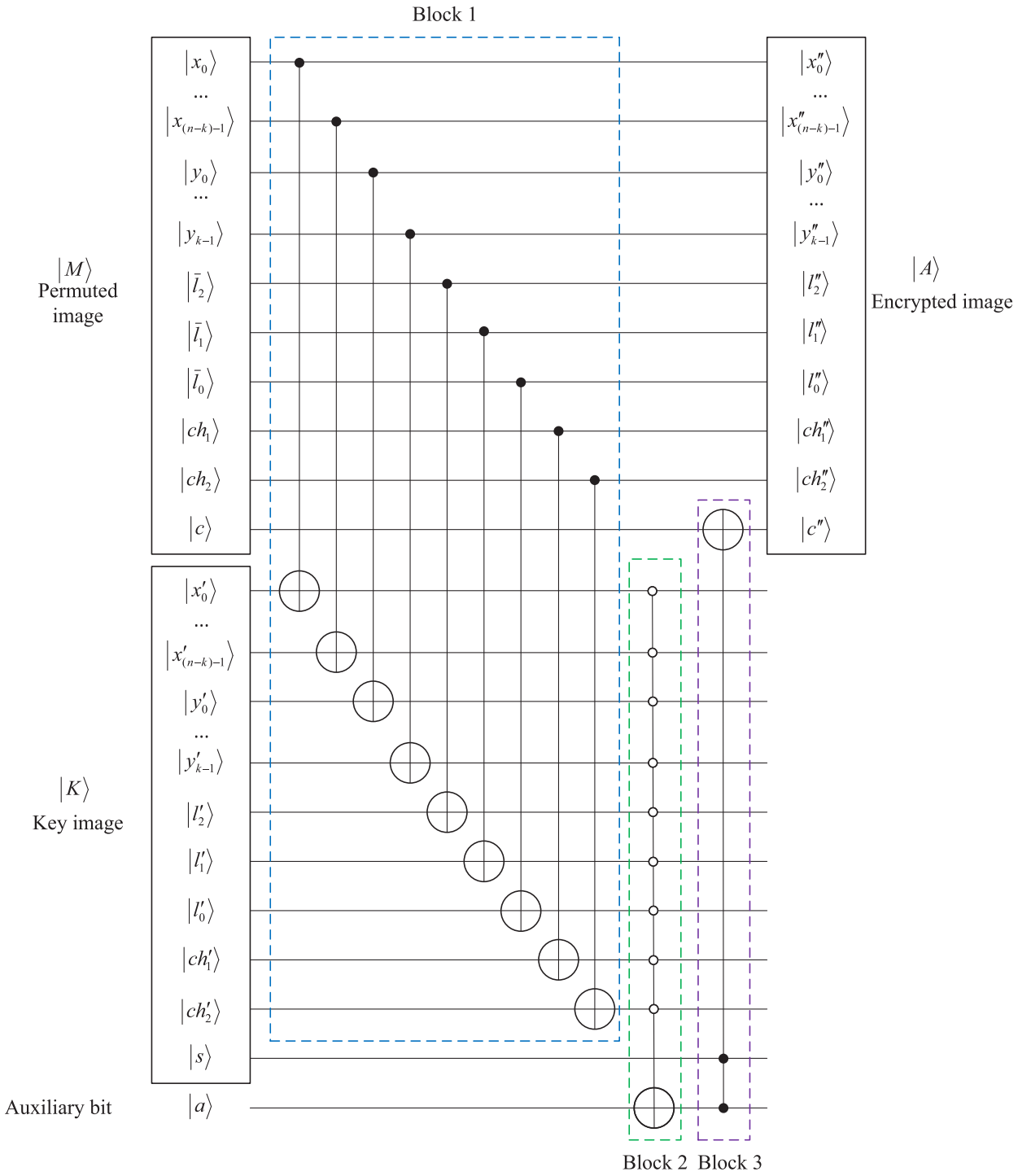


Fig. 10. XOR operation. By matching the positions of the key image $|K\rangle$ and the permuted image $|M\rangle$ one by one, the alignment operation is completed. Finally, the XOR operation of $|c\rangle$ is performed.

4. Simulation result and complexity analysis

To verify the effectiveness and performance of OCPBP, the simulation process of the encryption algorithm is implemented in MATLAB R2022b on a 64-bit computer using an Intel(R) Core(TM) i5-10210U CPU processor, 16.0 GB random access memory, and Windows 11 operating system. The keys used in the simulation include:

R-channel: μ_1=4.00, μ_2=3.90, $x_1(1)$=0.20, $x_2(1)$=0.90, λ_r=1015;
G-channel: μ_1=3.98, μ_2=3.99, $x_1(1)$=0.30, $x_2(1)$=0.90, λ_g=1015;
B-channel: μ_1=4.00, μ_2=3.97, $x_1(1)$=0.60, $x_2(1)$=0.90, λ_b=1015.

4.1. Encryption performance

In this subsection, the encryption performance of the proposed OCPBP is shown in Figs. 12 and 13. Usually, a uniform pixel distribution histogram [34] can effectively prevent an attacker from decrypting an image through statistical analysis. As shown in Figs. 12 and 13, the image encrypted by OCPBP is unable to reveal any features or content of the source image directly. Additionally, the histogram shows that the encrypted image's pixel distribution is uniform. Therefore, it can be verified that OCPBP has extensive adaptability in encrypting different

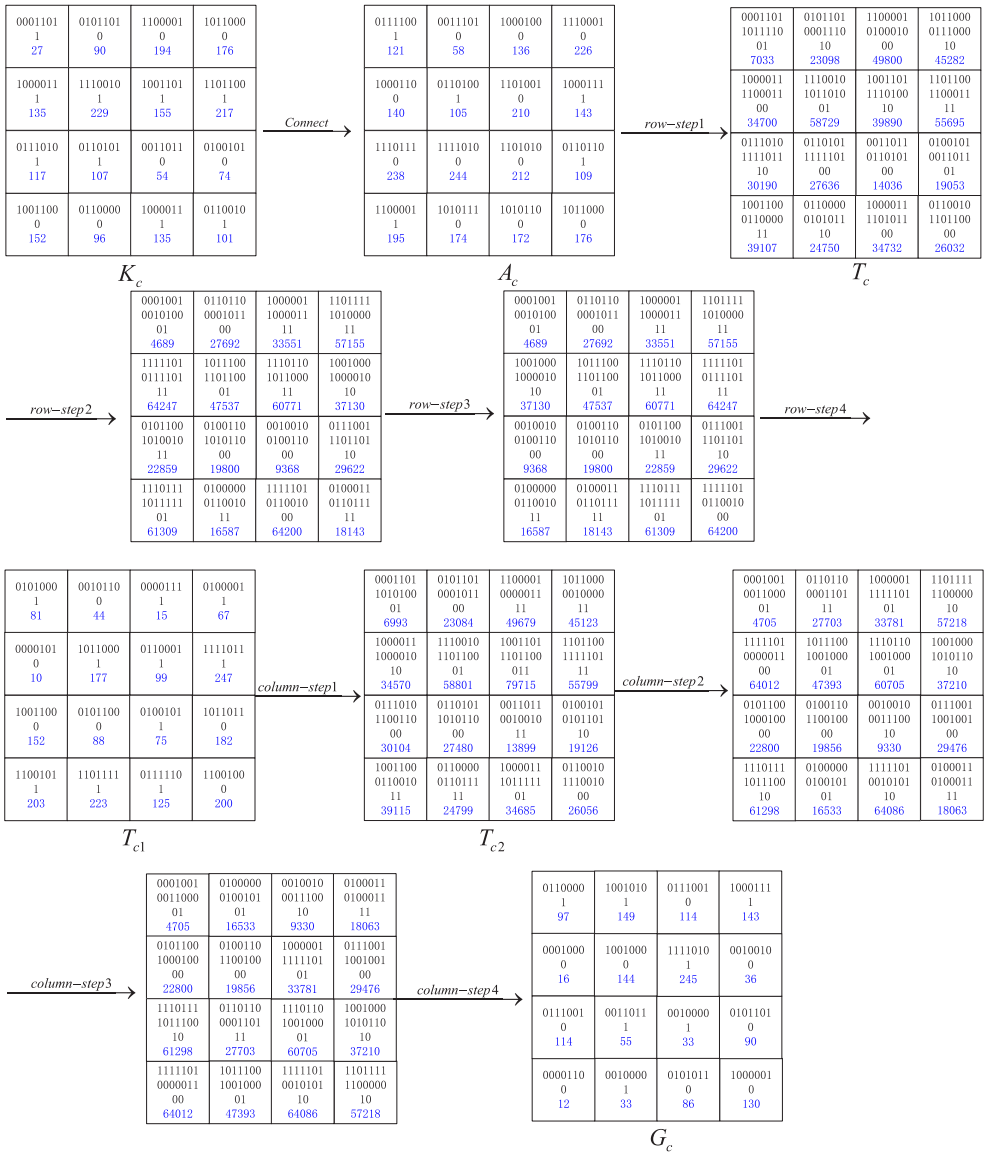


Fig. 11. Permutation operation in single channel image. According to the encryption process in Section 3.4, perform row encryption and column encryption on the sample image, where the blue number represents the color value of the current pixel, and the black number represents the binary color information of the value. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

types of images and can effectively resist attacks through statistical analysis.

4.2. The efficiency analysis in preparation circuit

In addition to the encryption performance, the memory consumption is another index to show the algorithm efficiency, and the smaller the number of quantum bits required for the quantum image preparation, the smaller the memory space required by the algorithm.

For a $2^{n-k} \times 2^k$ color image, the preparation process of the BRQI image requires n+6 qubits combining with n+3 auxiliary qubits, and the total number of qubits is 2n+9. In the proposed OCPBP, the preparation process requires n+5 qubits only combined with two auxiliary qubits, and the total number of qubits is n+8. Therefore, the proposed OCPBP has lower complexity. Moreover, to illustrate the effect of the auxiliary qubits clearer, for a $2^{n-k} \times 2^k = 2^{12}$ image with n=12, the original BRQI preparation circuit has a total of 33 qubits. In contrast, the optimized circuit requires only 20 qubits.

Table 2
The number of bits corresponds to the memory.

Approach	Quantum Bits	Memory size
OCPBP	20	17.28 MB
BRQI	33	138.24 GB

Note that each auxiliary qubit will significantly increase the required memory space. According to the memory space table of different quantum bits given by IBM laboratory, the 25 bits require 0.54 GB of memory. Each additional qubit doubles the memory size required. Therefore, 20 qubits require a memory space of $0.54 \times 2^{-5} \approx$17.28 MB, and 33 qubits require a memory space of $0.54 \times 2^8 \approx$138.24 GB. As shown in Table 2, BRQI requires a memory space of 138.24 GB, while OCPBP only requires 17.28 MB. Therefore, the optimized circuit can significantly reduce the memory and improve the preparation more obviously; it also verifies the efficiency of the proposed OCPBP.

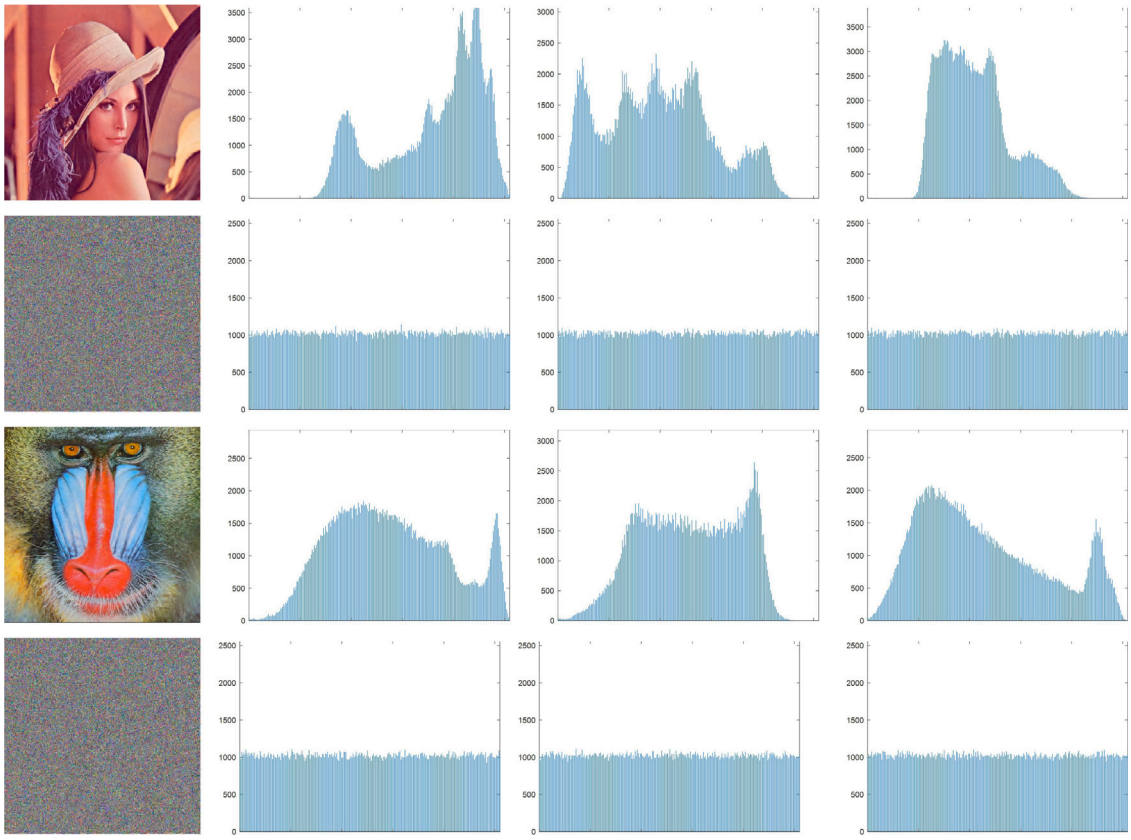


Fig. 12. RGB channel histogram of plain image and encrypted image of color images “Lena” and “Baboon”(512*512). The three histograms on the right side of each image are gray-scale histograms of R, G, and B channels.

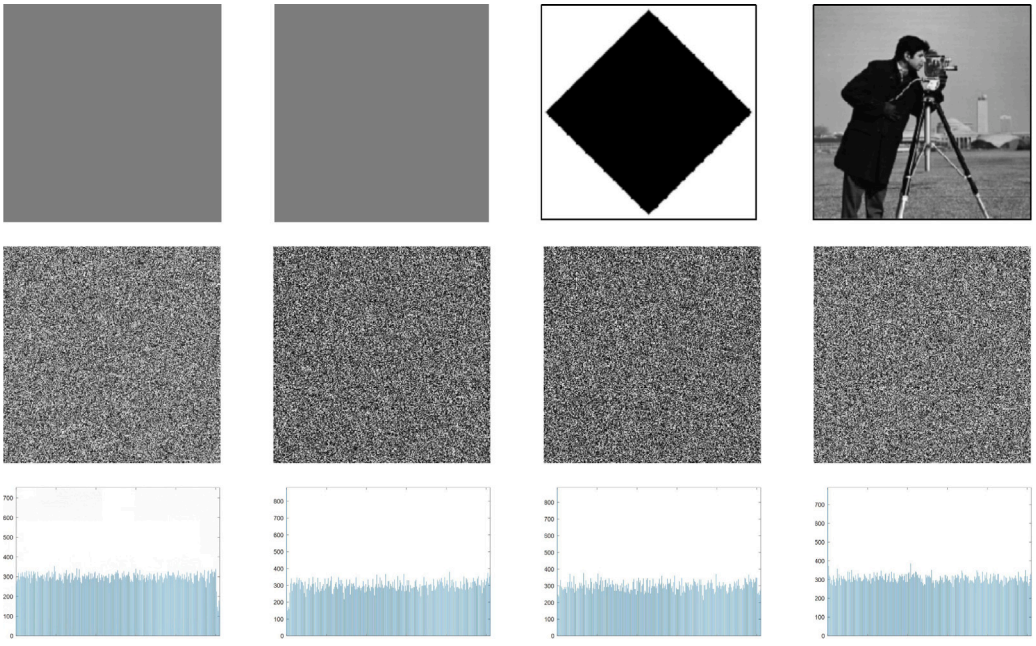


Fig. 13. Black image, white image, gray image and binary image and the corresponding encrypted image and its corresponding histogram.

Table 3
Complexity comparison.

Operation	OCPBP	Ref. [25]	Ref. [26]
permutation operation	$O(1)$	$O(1)$	$O(m)$
XOR operation	$O(n)$	$O(n+k)$	–

Table 4
Comparative analysis of encryption time for 512 × 512 'Lena'.

	OCPBP	Ref. [36]	Ref. [37]
Lena	2.236 s	19.14 s	7.305 s
Pepper s	2.288 s	19.97 s	6.748 s
Baboon	2.189 s	21.50 s	6.598 s

4.3. The complexity analysis of bit-plane-based permutation and XOR operation

Besides the complexity of optimization of the quantum image preparation, the complexity of the new bit-plane-based permutation operation proposed is also an essential component of OCPBP, which includes five Toffoli gates (30 CNOT gates) and three CNOT gates for even and odd pixels, respectively. Therefore, for a $2^{n-k} \times 2^k$ image, the permutation operation requires a total of 66 CNOT gates, and its time complexity is $O(1)$.

The complexity calculation of the XOR operation regards the CNOT gate as the primary quantum gate. Since a k-CNOT gate can be decomposed into $12k$-5 CNOT gate, a Toffoli gate is equivalent to 6 CNOT gates [35]. In Fig. 10, there are n+5 CNOT gates, one (n+5)-CNOT gate, and one Toffoli gate, which is equivalent to $13n$+66 CNOT gates, the algorithm complexity is $O(n)$.

Since the permutation in Section 3.4 does not involve the use of quantum gates, for a $2^{n-k} \times 2^k$ image, the OCPBP algorithm requires a total of $13n$+132 CNOT gate, i.e. the algorithm complexity is $O(n)$. The comparable results between the proposed OCPBP with Ref. [25] and Ref. [26] that also use the BRQI model to store images are shown in Table 3. It can be seen that the proposed OCPBP has a lower complexity.

4.4. Speed analysis

The speed of the encryption algorithm also reflects the level of encryption efficiency. Therefore, to analyze the time complexity of the proposed algorithm, we compare the encryption time of OCPBP with Ref. [36] and Ref. [37], and the results are as shown in Table 4. It can be seen encryption time of OCPBP is significantly shorter than Ref. [36] and Ref. [37]. Therefore, it is proved that OCPBP has high encryption efficiency.

5. Security analysis

Several security performance analyses are employed to demonstrate excellent security performance. Since there are few quantum encryption algorithms stored using the BRQI model, to verify the performance of OCPBP, we select the algorithm [25] that also uses the BRQI model and several chaos-based encryption algorithms [36–38] proposed in the past two years to illustrate the performance of the proposed OCPBP.

5.1. Key size analysis

High randomness in the key can guarantee high security for the encryption algorithm. The security of the encryption algorithm cannot be guaranteed if the key is poor randomness or the key space is small because these factors make the encryption key easily decipherable.

According to Ref. [39], the size of the security key space should be greater than 2^{100}. The key of OCPBP in the monochrome channel is: $KEY = \{\mu_1, \mu_2, x_1(1), x_2(1), \lambda\}$. Assuming that the precision of the non-integer key is 10^{-14}, and the precision of λ is 10^3, the key space size of OCPBP (gray-scale image) can reach $10^{14\times4} \times 10^3 = 10^{59} > 2^{100}$. The key space size for color images reaches 10^{177}. Additionally, we give the key space size of OCPBP and similar algorithms in Table 5. It can be seen in Table 5 that the OCPBP has a key space sufficiently large to fend off various brute force attacks.

Table 5
Key space comparison.

Algorithm	OCPBP	Ref. [36]	Ref. [38]	Ref. [37]
Key size	10^{177}	10^{112}	10^{84}	10^{35}

5.2. Correlation analysis

An excellent encryption algorithm must minimize the correlation between adjacent pixels in the encrypted image given the high correlation between adjacent pixels in an image [40]. The adjacent pixel correlation is calculated by

$$r_{xy} = \frac{\mathrm{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{22}$$

where x and y are two adjacent pixel values in the image, and the larger r_{xy} is, the stronger the correlation between adjacent pixels.

To test the correlation between adjacent pixels of the encrypted image, we select 30,000 pairs of adjacent pixel correlations from the plain image and encrypted images, compare them with the data obtained from four different encryption methods, and bold the better results (those with less correlation in the encrypted images). As shown in Table 6, the proposed OCPBP has 19 correlation coefficients that are the smallest among all compared algorithms, Ref. [38] has five correlation coefficients that are the smallest, Ref. [25] has two correlation coefficients that are the smallest, Ref. [36] has one correlation coefficient that is the smallest, while Ref. [37] have no minimum results. It can be concluded that our encryption algorithm has the best performance among all comparable algorithms.

5.3. Information entropy analysis

Information entropy [41] represents the degree of confusion of image information. The greater the entropy, the higher the degree of confusion, and the more difficult it is for the information to be cracked. For a random gray-scale image, the idea information entry is 8, which can be illustrated by

$$H(S) = -\sum_{i=0}^{2^N-1} p\left(s_i\right) \log_2 p\left(s_i\right), \tag{23}$$

where S is the gray-scale set, and $p\left(s_i\right)$ represents the frequency of the i-th color. Table 7 gives the information entropy results of the three RGB images. We bold the results that are closer to 8 in each row. As shown in Table 7, the information entropy of OCPBP is closer to 8 than other comparable encryption algorithms, which verifies the better uncertainty of the OCPBP.

5.4. Differential attack analysis

Along with having superior encryption results, a good encryption method should be able to survive differential attacks. A differential attack is a selected plain text attack, according to Ref. [42], even a one-pixel difference in the plain image must result in a significant difference in the encryption result. Here, we test the OCPBP's ability to fend off differential attacks using the Number of Pixels Change Rate ($NPCR$)

Table 6
Correlation analysis of color image encryption.

Image	Direction	Plain image	OCPBP	Ref. [25]	Ref. [36]	Ref. [38]	Ref. [37]
Lena.R	Horizontal	0.9893	0.0023	−0.0014	**−0.0006**	−0.0025	0.0064
	Vertical	0.9798	**0.0008**	−0.0047	−0.00497	0.0021	0.0160
	Diagonal	0.9698	**−0.0004**	0.0018	0.0070	0.0026	−0.0026
Lena.G	Horizontal	0.9828	**−0.0006**	−0.0010	0.0025	−0.0014	0.0009
	Vertical	0.9689	**0.0005**	−0.0092	−0.0051	0.0016	0.0034
	Diagonal	0.9557	0.0003	−0.0011	0.0020	**0.00001**	0.0125
Lena.B	Horizontal	0.9575	**−0.0001**	−0.0043	0.0046	−0.0006	0.0091
	Vertical	0.9329	**0.0004**	0.0006	0.0019	0.0019	−0.0045
	Diagonal	0.9180	0.0026	−0.0012	0.0047	**−0.0001**	−0.0090
Peppers.R	Horizontal	0.9663	**0.0009**	0.0003	−0.0011	−0.0022	−0.0145
	Vertical	0.9636	**0.0005**	0.0012	0.0015	−0.0007	0.0071
	Diagonal	0.9563	**0.0009**	−0.0012	0.0066	−0.0020	0.0200
Peppers.G	Horizontal	0.9820	−0.0015	−0.0010	−0.0015	**0.0006**	0.0149
	Vertical	0.9815	**0.0010**	0.0024	0.0059	−0.0016	0.0136
	Diagonal	0.9693	0.0020	0.0008	−0.0028	**−0.0007**	0.0067
Peppers.B	Horizontal	0.9662	**−0.0003**	−0.0050	−0.0042	−0.0006	0.0393
	Vertical	0.9665	**−0.0008**	0.0246	0.0037	−0.0034	−0.0019
	Diagonal	0.9477	**−0.0002**	−0.0110	0.0025	0.0022	−0.0038
Baboon.R	Horizontal	0.9893	−0.0007	**−0.0001**	0.0052	0.0006	−0.0213
	Vertical	0.9798	**0.0002**	0.0025	−0.0042	−0.0003	0.0072
	Diagonal	0.9698	**0.0002**	0.0007	0.0008	0.0008	0.0011
Baboon.G	Horizontal	0.9828	0.0006	−0.0017	0.0047	**0.0004**	0.0126
	Vertical	0.9689	**0.0003**	−0.00113	−0.0071	0.0024	0.0120
	Diagonal	0.9557	−0.0006	**0.0002**	0.0009	−0.0044	−0.0133
Baboon.B	Horizontal	0.9575	**−0.0001**	−0.0019	−0.0003	−0.0003	−0.0102
	Vertical	0.9329	**0.0008**	0.0024	−0.0013	0.0021	0.0015
	Diagonal	0.9180	**0.0001**	−0.0009	−0.0028	0.0019	0.0025

Table 7
Information entropy of the images.

Image	Color	Plain	OCPBP	Ref. [25]	Ref. [36]	Ref. [38]	Ref. [37]
Lena	R	7.2525	**7.9994**	7.9938	7.9966	7.9992	7.9974
	G	7.5940	**7.9994**	7.9951	7.9972	7.9993	7.9976
	B	6.6984	**7.9994**	7.9952	7.9972	7.9993	7.9974
Peppers	R	7.3388	7.9992	7.9970	7.9970	7.9992	**7.9993**
	G	7.4963	**7.9993**	7.9968	7.9965	7.9992	**7.9993**
	B	7.0583	**7.9993**	7.9986	7.9962	**7.9993**	**7.9993**
Baboon	R	7.7064	**7.9993**	7.9968	7.9972	**7.9993**	**7.9993**
	G	7.4740	**7.9993**	7.9983	7.9969	**7.9993**	**7.9993**
	B	7.7522	**7.9994**	7.9953	7.9971	7.9993	7.9993

and Unified Average Changing Intensity ($UACI$) metrics [43,44]. They are characterized by

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%, \tag{24}$$

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\%, \tag{25}$$

where C_1 and C_2 are two encrypted images with the same size $W \times H$. If $C_1(i,j) \neq C_2(i,j)$ then $D(i,j) = 1$, else $D(i,j) = 0$. Under ideal conditions, the $NPCR$ and $UACI$ value is 99.6094% and 33.4635%, respectively [45]. Table 8 shows the $NPCR$ and $UACI$ results of the encrypted image of OCPBP and the comparable algorithms. The proposed OCPBP is highly sensitive to plain images; even the slightest modification will result in completely different encrypted images, making it more resistant to differential attacks.

5.5. Anti-noise attack analysis

Noise attacks commonly occur during communication. To demonstrate the robustness of OCPBP, this paper utilizes the "Lena" image as the test subject. Gaussian noise attacks are conducted with strengths of 0.0001, 0.0003, and 0.0005, while Salt and pepper noise attacks are applied at strengths of 0.01, 0.03, and 0.05. The corresponding decrypted images are displayed in Fig. 14. It is evident that even when the encrypted images are subjected to various noise attacks, they can still be successfully decrypted with an acceptable visual quality. However, as the intensity of the attack increases, the image restoration effect weakens. Consequently, we conducted further tests with Salt and pepper noise, increasing the intensity to 0.1, 0.3, and 0.5. The decryption effects are depicted in Fig. 14(c). As the noise intensity reaches 0.5, the image's decryption quality deteriorates but still maintains an acceptable visual standard.

Fig. 14. Anti-noise attack analysis. (a) the decrypted images with Gaussian noise attacks with strengths of 0.0001, 0.0003, and 0.0005. (b) the decrypted images with Salt and pepper noise with strengths of 0.01, 0.03, and 0.05. (c) the decrypted images with Salt and pepper noise with strengths of 0.1, 0.3, and 0.5.

Table 8
The $NPCR$ (%) and $UACI$ (%) of the images.

Image	$NPCR.R$	$NPCR.G$	$NPCR.B$	Mean	$UACI.R$	$UACI.G$	$UACI.B$	Mean
OCPBP.Lena	99.5963	99.6151	99.6054	99.6056	33.5012	33.4322	33.4598	33.4644
Ref. [36]	99.6535	99.5770	99.6560	99.6225	33.4943	33.5117	33.5901	33.5320
Ref. [38]	99.6100	99.6100	99.5900	99.6033	30.2400	30.9700	32.300	31.1700
Ref. [37]	99.6170	99.5895	99.6231	99.6099	33.4555	33.4496	33.5901	33.4984
OCPBP.Baboon	99.6062	99.6121	99.6087	99.6090	33.4582	33.4701	33.4676	33.4653
Ref. [36]	99.5895	99.4660	99.5712	99.5422	33.4593	33.4282	33.4780	33.4614
Ref. [38]	99.6200	99.6200	99.6100	99.6167	29.9500	28.6300	31.2000	29.9267
Ref. [37]	99.6014	99.6254	99.6082	99.6117	33.4578	33.4415	33.4102	33.4365
OCPBP.Peppers	99.6132	99.6011	99.6085	99.6076	33.4679	33.4581	33.5050	33.4770
Ref. [36]	99.4819	99.5876	99.4827	99.5174	33.4582	33.5087	33.4378	33.4682
Ref. [38]	99.5900	99.6200	99.6000	99.6033	28.9800	33.9600	33.7300	32.2233
Ref. [37]	99.5987	99.6082	99.6044	99.6038	33.4899	33.4060	33.4772	33.4577

Table 9
PSNR (dB) results.

Gaussian	OCPBP	Ref. [37]	Salt pepper	OCPBP	Ref. [37]
0.0001	19.0856	14.8527	0.01	33.3325	28.7554
0.0003	19.0932	14.8631	0.03	28.3793	23.9007
0.0005	19.0598	14.8633	0.05	26.2546	21.7124
Salt pepper	OCPBP	Ref. [37]	Salt pepper	OCPBP	Ref. [37]
0.002-R	39.2553	34.8422	0.005-R	35.8309	31.0633
0.002-G	40.3721	35.5262	0.005-G	36.4796	31.7893
0.002-B	41.7669	36.7838	0.005-B	37.6775	32.4451

Simultaneously, we compute the Peak Signal-to-Noise Ratio ($PSNR$) between the plain and encrypted images. $PSNR$ serves as a widely accepted metric for quantitatively assessing signal reconstruction quality, effectively emphasizing the resilience of the proposed method. For monochrome images I and K with dimensions of $m \times n$, if one image approximates the noise of the other, their mean square error can be defined as

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\|I(i,j) - K(i,j)\|^2, \tag{26}$$

$$PSNR = 10\log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right), \tag{27}$$

where MAX_I is the maximum value of pixel color if each sampling point is represented by 8 bits, then $MAX_I = 255$, the results of $PSNR$ are as given in Table 9. According to the $PSNR$ comparison between OCPBP and the algorithm in Ref. [37], the decrypted results of the attacked image by OCPBP have higher $PSNR$ values for the decrypted images. This finding underscores the superior quality of reconstruction offered by OCPBP compared to the competing algorithms.

5.6. Malicious cropping attack resistance

Encrypted images are typically easy to obtain in public, and attackers could pretend to change or crop some of the information in the encrypted image, resulting in some information loss and making the image more challenging to decrypt. Therefore, the cropped encrypted images may be made so on purpose. Here, we demonstrate the OCPBP decrypted image that has been cropped; the outcome is as displayed in Fig. 15. As shown in Fig. 15, even if the encrypted image is partially missing, its decrypted image still has an acceptable visual reconstruction of the original image's content. It also verifies the robustness of OCPBP.

5.7. Deep learning attack resistance

Generative Adversarial Networks (GAN) [46] is an unsupervised deep learning model. The Model produces a fairly good output through game learning between the Generative Model and the Discriminative Model, which can be illustrated by

- Generator: for a given random vector values as inputs, this network tries to generate data with the same structure as the training data.
- Discriminator: for a given data batch containing observations from the training data and generated data from the Generator, this network attempts to divide the observations as "real" or "generated".

We train GAN using the Minist dataset to test OCPBP's resistance to deep learning attacks. The training procedure specifies the Adam optimizer, learning rate = 0.0001, gradient attenuation factor = 0.5, gradient square attenuation factor = 0.999. First, according to the classical GAN algorithm, we input six random noise images and use Minist data set the image as the expected output with 500 round confrontation training. After the trained network inputs 6 noisy images, it can generate images similar to the data set. Subsequently, we use OCPBP encrypted six images in Minist. As shown in Fig. 16(a–b). Then, the encrypted images are used as the input of GAN, and the plain images are used as the expected output of training. After 500 rounds of training with the same model parameters, the generated images are shown in Fig. 16(d). Compared with Fig. 16(c) and (d), the GAN cannot establish an acceptable visual connection with the original image, and the proposed OCPBP can resist the deep learning attack.

6. Comparison

Through the simulation analysis in Section 4 and security analysis in Section 5, we can conclude that OCPBP has higher encryption efficiency and security than the compared algorithms. To illustrate the performance of OCPBP intuitionally, Table 10 provides a comprehensive overview of OCPBP and the comparable algorithms. It shows that OCPBP exhibits low complexity, low correlation between adjacent pixels, high information entropy, higher key space, and high NPCR and UACI values. In addition, OCPBP can effectively resist differential attacks, noise attacks, Malicious cropping attacks, and deep learning attacks. All these verify the deficiency and performance of OCPBP.

7. Conclusions

This paper introduces the OCPBP quantum encryption algorithm, which incorporates an optimized quantum circuit and a parity bit-plane permutation. It effectively addresses the issue of non-random keys observed in existing BRQI-based encryption methods. The bit-plane-based permutation operation, in conjunction with the row-column-based permutation operation, not only alters the color of individual pixels but also shuffles their positions, enhancing overall security.

Furthermore, our optimization technique for the BRQI image preparation circuit significantly reduces both complexity and memory requirements. Specifically, it reduces the number of auxiliary qubits

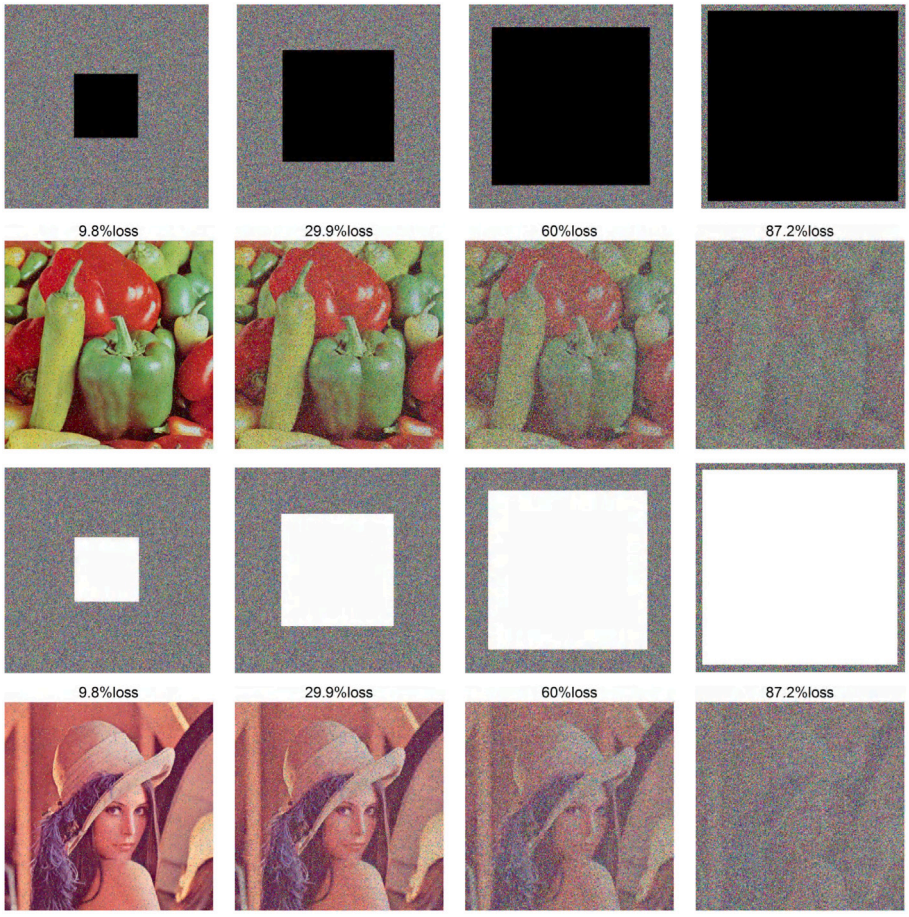


Fig. 15. Decryption effect after the encrypted image is cut and lost. From left to right, there is a pixel loss of 9.8%, 29.9%, 60%, and 87.2% in both the ciphertext image and its decrypted image.

Table 10
Comparison between OCPBP and similar algorithms.

Algorithm	Complexity	Encryption time	Key space	Correlation	Information entropy	$NPCR$ (%) and $UACI$(%)
OCPBP	O(n)	2.236 s	10^{177}	Low	High	High
Ref. [25]	O($n+k$)	–	–	Medium	Low	–
Ref. [26]	O(m)	–	–	Medium	Low	–
Ref. [36]	–	19.14 s	10^{112}	High	Low	Medium
Ref. [37]	–	7.305 s	10^{84}	High	Medium	Low
Ref. [38]	–	–	10^{35}	Medium	Medium	Low

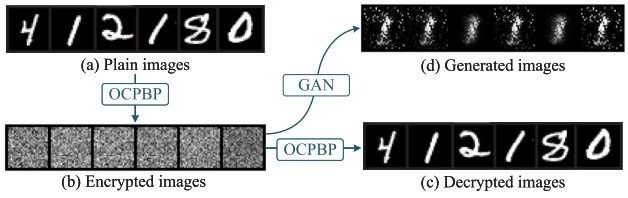


Fig. 16. The generated image using GAN. (a) Plain images (b) The input encrypted images using OCPBP (c) Decrypted images (d) Generated images (OCPBP).

needed from n+3 to just 2 for (n+5)-CNOT gate preparation. The simulation and contrast experiments illustrate that OCPBP exhibits a favorable combination of low complexity and robust security. It is also appropriate for gray-scale, color, and various types of images. In the future, we plan to extend the encryption algorithm to scramble the encryption of color channels based on the properties of the BRQI model.

CRediT authorship contribution statement

Jinwen He: Data curation, Methodology, Visualization, Writing – original draft. **Hegui Zhu:** Conceptualization, Formal analysis, Methodology, Project administration, Supervision, Writing – review & editing. **Xv Zhou:** Data curation, Investigation.

Declaration of competing interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript.

Data availability

Data will be made available on request.

Acknowledgments

This work was funded by the Natural Science Foundation of Liaoning Province, China (No. 2020-MS-080), the Fundamental Research Funds for the Central Universities (No. N2224005-4), and the National Students Innovation and Entrepreneurship Training Program (No. 220033).

References

[1] Zhang Q, Han J. A novel color image encryption algorithm based on image hashing, 6D hyperchaotic and DNA coding. Multimedia Tools Appl 2021;80(9):13841–64. http://dx.doi.org/10.1007/s11042-020-10437-z.
[2] Cao C, Sun K, Liu W. A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map. Signal Process 2018;143:122–33. http://dx.doi.org/10.1016/j.sigpro.2017.08.020.
[3] Borujeni SE, Eshghi M. Chaotic image encryption system using phase-magnitude transformation and pixel substitution. Telecommun Syst 2013;52(2):525–37. http://dx.doi.org/10.1007/s11235-011-9458-8.
[4] Ma Y, Li C, Ou B. Cryptanalysis of an image block encryption algorithm based on chaotic maps. J Inf Secur Appl 2020;54:102566. http://dx.doi.org/10.1016/j.jisa.2020.102566.
[5] Cao C, Sun K, Liu W. A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map. Signal Process 2018;143:122–33. http://dx.doi.org/10.1016/j.sigpro.2017.08.020.
[6] Lin R, Liu S, Jiang J, Li S, Li C, Kuo C-CJ. Recovering sign bits of DCT coefficients in digital images as an optimization problem. J Vis Commun Image Represent 2024;99:104045. http://dx.doi.org/10.1016/j.jvcir.2023.104045.
[7] Zhou S, Deng X, Li C, Liu Y, Jiang H. Recognition-oriented image compressive sensing with deep learning. IEEE Trans Multimed 2023;25:2022–32. http://dx.doi.org/10.1109/TMM.2022.3142952.
[8] Han N, Zeng Y, Shi C, Xiao G, Chen H, Chen J. Bic-net: learning efficient spatio-temporal relation for text-video retrieval. ACM Trans Multimed Comput Commun Appl 2023;20(3):1–21.
[9] Schaller R. Moore's law: Past, present, and future. IEEE Spectr 1997;34(6):52+. http://dx.doi.org/10.1109/6.591665.
[10] Keyes R. Miniaturization of electronics and its limits (Reprinted from IBM Journal of Research and Development, vol 32, 1988). IBM J Res Dev 2000;44(1–2):84–8. http://dx.doi.org/10.1147/rd.441.0084.
[11] Henzinger M, Rubinfeld R. Twenty-eighth annual ACM symposium on the theory of computing - editors' foreword. J Comput System Sci 1999;58(1):100.
[12] Shi J, Chen S, Lu Y, Feng Y, Shi R, Yang Y, et al. An approach to cryptography based on continuous-variable quantum neural network. Sci Rep 2020;10(1). http://dx.doi.org/10.1038/s41598-020-58928-1.
[13] Shor P. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J Comput 1997;26(5):1484–509. http://dx.doi.org/10.1137/S0097539795293172.
[14] Le PQ, Dong F, Hirota K. A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. Quantum Inf Process 2011;10(1):63–84. http://dx.doi.org/10.1007/s11128-010-0177-y.
[15] Zhang Y, Lu K, Gao Y, Wang M. NEQR: a novel enhanced quantum representation of digital images. Quantum Inf Process 2013;12(8):2833–60. http://dx.doi.org/10.1007/s11128-013-0567-z.
[16] Jiang N, Wang J, Mu Y. Quantum image scaling up based on nearest-neighbor interpolation with integer scaling ratio. Quantum Inf Process 2015;14(11):4001–26. http://dx.doi.org/10.1007/s11128-015-1099-5.
[17] Jiang N, Wang J, Mu Y. Quantum image scaling up based on nearest-neighbor interpolation with integer scaling ratio. Quantum Inf Process 2015;14(11):4001–26. http://dx.doi.org/10.1007/s11128-015-1099-5.
[18] Li H-S, Chen X, Xia H, Liang Y, Zhou Z. A quantum image representation based on bitplanes. IEEE Access 2018;6:62396–404. http://dx.doi.org/10.1109/ACCESS.2018.2871691.
[19] Akhshani A, Akhavan A, Lim SC, Hassan Z. An image encryption scheme based on quantum logistic map. Commun Nonlinear Sci Numer Simul 2012;17(12):4653–61. http://dx.doi.org/10.1016/j.cnsns.2012.05.033.
[20] Yang Y-G, Jia X, Sun S-J, Pan Q-X. Quantum cryptographic algorithm for color images using quantum Fourier transform and double random-phase encoding. Inform Sci 2014;277:445–57. http://dx.doi.org/10.1016/j.ins.2014.02.124.
[21] Yang Y-G, Xia J, Jia X, Zhang H. Novel image encryption/decryption based on quantum Fourier transform and double phase encoding. Quantum Inf Process 2013;12(11):3477–93. http://dx.doi.org/10.1007/s11128-013-0612-y.
[22] Zhang J, Huo D. Image encryption algorithm based on quantum chaotic map and DNA coding. Multimedia Tools Appl 2019;78(11):15605–21. http://dx.doi.org/10.1007/s11042-018-6973-6.
[23] Zhou N-R, Huang L-X, Gong L-H, Zeng Q-W. Novel quantum image compression and encryption algorithm based on dqwt and 3d hyper-chaotic henon map. Quantum Inf Process 2020;19(9). http://dx.doi.org/10.1007/s11128-020-02794-3.
[24] Ye G, Wu H, Jiao K, Mei D. Asymmetric image encryption scheme based on the Quantum logistic map and cyclic modulo diffusion. Math Biosci Eng 2021;18(5):5427–48. http://dx.doi.org/10.3934/mbe.2021275.
[25] Khorrampanah M, Houshmand M, Heravi MML. New method to encrypt RGB images using quantum computing. Opt Quantum Electron 2022;54(4). http://dx.doi.org/10.1007/s11082-022-03581-3.
[26] Heidari S, Naseri M, Nagata K. Quantum selective encryption for medical images. Internat J Theoret Phys 2019;58(11):3908–26. http://dx.doi.org/10.1007/s10773-019-04258-6.
[27] Yang G, Song X, Hung WNN, Xie F, Perkowski MA. Group theory based synthesis of binary reversible circuits. In: Theory and applications of models of computation, proceedings. 2006, p. 365–74.
[28] Suzhen Yuan YL. Quantum image processing and its implementation. Beijing: Science Press; 2019, p. 30–50.
[29] Hua Z, Zhang Y, Bao H, Huang H, Zhou Y. n-dimensional polynomial chaotic system with applications. IEEE Trans Circuits Syst I-Regul Pap 2022;69(2):784–97. http://dx.doi.org/10.1109/TCSI.2021.3117865.
[30] Bao B, Rong K, Li H, Li K, Hua Z, Zhang X. Memristor-coupled logistic hyperchaotic map. IEEE Trans Circuits Syst II-Express Briefs 2021;68(8):2992–6. http://dx.doi.org/10.1109/TCSII.2021.3072393.
[31] Liu J, Tang S, Lian J, Ma Y, Zhang X. A novel fourth order chaotic system and its algorithm for medical image encryption. Multidimens Syst Signal Process 2019;30(4):1637–57. http://dx.doi.org/10.1007/s11045-018-0622-0.
[32] Zhu H, Ge J, Qi W, Zhang X, Lu X. Dynamic analysis and image encryption application of a sinusoidal-polynomial composite chaotic system. Math Comput Simulation 2022;198:188–210. http://dx.doi.org/10.1016/j.matcom.2022.02.029.
[33] Zhu H, Dai L, Liu Y, Wu L. A three-dimensional bit-level image encryption algorithm with Rubik's cube method. Math Comput Simulation 2021;185:754–70. http://dx.doi.org/10.1016/j.matcom.2021.02.009.
[34] Li B, Liao X, Jiang Y. A novel image encryption scheme based on logistic map and dynatomic modular curve. Multimedia Tools Appl 2018;77(7):8911–38. http://dx.doi.org/10.1007/s11042-017-4786-7.
[35] Shende VV, Markov IL. On the Cnot-cost of Toffoli Gates. Quantum Inf Comput 2009;9(5–6):461–86.
[36] Wang X, Su Y, Luo C, Nian F, Teng L. Color image encryption algorithm based on hyperchaotic system and improved quantum revolving gate. Multimedia Tools Appl 2022;81(10):13845–65. http://dx.doi.org/10.1007/s11042-022-12220-8.
[37] Hosny KM, Kamal ST, Darwish MM. A color image encryption technique using block scrambling and chaos. Multimedia Tools Appl 2022;81(1):505–25. http://dx.doi.org/10.1007/s11042-021-11384-z.
[38] Alghafis A, Munir N, Khan M, Hussain I. An encryption scheme based on discrete quantum map and continuous chaotic system. Internat J Theoret Phys 2020;59(4):1227–40. http://dx.doi.org/10.1007/s10773-020-04402-7.
[39] Wang X, Yang J. A privacy image encryption algorithm based on piecewise coupled map lattice with multi dynamic coupling coefficient. Inform Sci 2021;569:217–40. http://dx.doi.org/10.1016/j.ins.2021.04.013.
[40] Wang X, Yang J. A privacy image encryption algorithm based on piecewise coupled map lattice with multi dynamic coupling coefficient. Inform Sci 2021;569:217–40. http://dx.doi.org/10.1016/j.ins.2021.04.013.
[41] Abd El-Latif AA, Niu X, Amin M. A new image cipher in time and frequency domains. Opt Commun 2012;285(21–22):4241–51. http://dx.doi.org/10.1016/j.optcom.2012.06.041.
[42] Wang X, Gao S. Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network. Inform Sci 2020;539:195–214. http://dx.doi.org/10.1016/j.ins.2020.06.030.
[43] Ahmad J, Khan MA, Hwang SO, Khan JS. A compression sensing and noise-tolerant image encryption scheme based on chaotic maps and orthogonal matrices. Neural Comput Appl 2017;28(1):S953–67. http://dx.doi.org/10.1007/s00521-016-2405-6.
[44] Abd-El-Hafiz SK, AbdElHaleem SH, Radwan AG. Novel permutation measures for image encryption algorithms. Opt Lasers Eng 2016;85:72–83. http://dx.doi.org/10.1016/j.optlaseng.2016.04.023.
[45] Wang X-Y, Li Z-M. A color image encryption algorithm based on Hopfield chaotic neural network. Opt Lasers Eng 2019;115:107–18. http://dx.doi.org/10.1016/j.optlaseng.2018.11.010.
[46] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. Commun ACM 2020;63(11):139–44. http://dx.doi.org/10.1145/3422622.