



A dual quantum image scrambling method

Shahrokh Heidari¹ · Matin Vafaei² · Monireh Houshmand² ·
Narges Tabatabaei-Mashadi²

Received: 4 February 2018 / Accepted: 9 November 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Considering blindness in information security, image scrambling method is defined as a procedure by which an image is turned into an absolutely different meaningless image through a reversible transformation. By scrambling an image, the ability to resist against unauthorized attacks and accordingly, augmenting security can be obtained effectively. In this approach, a quantum representation of a digital scrambling algorithm is investigated for quantum NCQI color images. Experimental consequences encompassing histogram diagram, entropy rate, correlation coefficient and number of pixels change ratio, which are analyzed in MATLAB environment, indicate a good performance, showing that the proposed method is much more secure and applicable than the previous one currently found in the literature related to the quantum image disordering methods.

Keywords Quantum image processing · Image scrambling · NCQI color image

1 Introduction

Affecting human's daily tasks considerably, secure communication seems to be one of the most noticeable issues. Undoubtedly, when people have more secure channels to communicate, more confidence and reliance they can have and consequently, they are able to send their important information through a secure way, in which the interlopers would be unable to distinguish the information.

✉ Monireh Houshmand
M.houshmand@imamreza.ac.ir

Shahrokh Heidari
Heidari.shahrokh@iauksh.ac.ir

¹ Young Researchers and Elite Club, Kermanshah Branch, Islamic Azad University, Kermanshah, Iran

² Department of Electrical Engineering, Imam Reza International University, Mashhad, Iran

When a kind of information such as image data is sent through a public channel, attackers may have easy access to them. Inevitably, we should have some useful techniques to impede such an illegal access. Generally, there are two main methods by which an image transmission could be culminated with security: image steganography and image watermarking, being defined as follows. Image steganography is the art science of concealing data in a cover image so as to send to the receiver, in which nobody except him/her can discern the existence of the hidden information [1]. Image watermarking procedure is a technique for embedding the indiscernible signal such as the owner identification into an image with the aim of copyright protection [2]. The most integral part of these techniques, as an image preprocessing method, is the image scrambling algorithm accounting for that the hidden image data before sending to the receiver are turned into a disorder image for enhancing the security. As regards this situation, even if the way of concealing the obscure data could be divulged and the attackers could extract the information, the only feat would be just having a meaningless image. Hence, the receiver who knows how the secret image have been scrambled can easily elicit the meaningful original image. As a matter of the fact, image scrambling brings about that illegal operations such as forgery, modification and duplication become significantly dwindled.

By the advent of quantum information theory and most notably its applications, this field of study has been considered by researchers, receiving more and more attention than ever before. As one of these applications, quantum image processing (QImP) is defined as using the quantum computing technologies to store, manipulate, and recover digital images to satisfy various goals [3]. The first essential object to consider in QImP, as a preliminary requirement, is to have a quantum representation of digital image. So, a variety of techniques have been investigated in order to encode a digital image by qubits [4–12]. Interestingly, mainly due to the expansion of these methods, a series of significant theoretical improvements have been proposed in this field [13–31].

The quantum image scrambling method is defined as implementing an algorithm on one of the quantum representations of digital images in order to turn a meaningful quantum image into a meaningless one by which the security can be enhanced dramatically. In 2014, Jiang et.al suggested a quantum image scrambling technique utilized Arnold and Fibonacci transition [32], introducing their approach based on the flexible representation for quantum images (FRQI) [6]. Subsequently, they proposed another quantum image scrambling technique called quantum Hilbert image scrambling [33], implementing an algorithm of Hilbert scanning matrix in order to scramble a FRQI image. Afterward, they tried to improve their first method [32] and by simplifying the old version as well as decreasing its complexity, a new approach was fulfilled [34]. In 2015, Zhou et. al investigated a quantum image scrambling method [35] by disordering bit-plane of pixels in which an NEQR image [8] was manipulated to produce a meaningless quantum image.

In this contribution, a dual quantum scrambling method for NCQI images introduced in Ref. [12] is investigated in which two algorithms are applied to a quantum image. The first algorithm manipulates bit-plane of the pixels and the second one changes the order of the pixels. It is worth pointing out that these algorithms are not executed simultaneously and the output of the first algorithm is considered as the

input of the second one and the final output would be what can be called as a quantum scrambled image.

The rest of this paper is organized as follows. A brief required explanation pertained to how a digital image can be illustrated as an NCQI image is depicted in Sect. 2. Next, our proposed method by the aim of performing quantum image scrambling is investigated in Sect. 3. The simulation-based experiments and analysis are given in Sect. 4. Then the comparison of our proposed method with a pertinent studied work is derived in Sect. 5. Finally, the conclusion is discussed in Sect. 6.

2 A novel quantum representation for color digital images (NCQI)

Inspired by the method named novel enhanced quantum representation (NEQR) [8], NCQI [12] was proposed by Sang et. al to store and process colored RGB digital images on quantum computers. In this type of quantum images, the color information is encoded in three channels (Red–Green–Blue: RGB). This method employs two entangled qubit sequences so that the information of the color and the position of each pixel can be encoded by qubits. The first sequence is used to encode the red, green and blue values of the color related to each pixel and the second sequence encodes the information about the position of the pixel. Suppose that the range of each channel is 2^q . The presentation of a $2^n \times 2^n$ RGB image can be shown as:

$$|I\rangle = \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} |C_{\text{RGB}_i}\rangle \otimes |i\rangle, \quad (1)$$

where \otimes represents the tensor product and the state $|C_{\text{RGB}_i}\rangle$ is used to encode the red, green and blue channels information of the pixel i , which can be defined as follows:

$$|C_{\text{RGB}_i}\rangle = |R_i\rangle |G_i\rangle |B_i\rangle. \quad (2)$$

Here, $|R_i\rangle$, $|G_i\rangle$ and $|B_i\rangle$ depict the information of the channels.

$$\begin{aligned} |R_i\rangle &= |r_i^{q-1} r_i^{q-2} \dots r_i^1 r_i^0\rangle, \\ |G_i\rangle &= |g_i^{q-1} g_i^{q-2} \dots g_i^1 g_i^0\rangle, \\ |B_i\rangle &= |b_i^{q-1} b_i^{q-2} \dots b_i^1 b_i^0\rangle, \\ r_i^k, g_i^k, b_i^k &\in \{0, 1\} \end{aligned}$$

and also $|i\rangle$ is shown as below.

$$|i\rangle = |y\rangle |x\rangle = |y_{n-1}, y_{n-2} \dots y_0\rangle |x_{n-1}, x_{n-2} \dots x_0\rangle, y_j x_j \in \{0, 1\}, \quad (3)$$

where $|i\rangle$ encodes the first n -qubit $|y_{n-1}\rangle, |y_{n-2}\rangle, \dots, |y_0\rangle$ along the vertical location and the second n -qubit $|x_{n-1}\rangle, |x_{n-2}\rangle, \dots, |x_0\rangle$ along the horizontal axis.

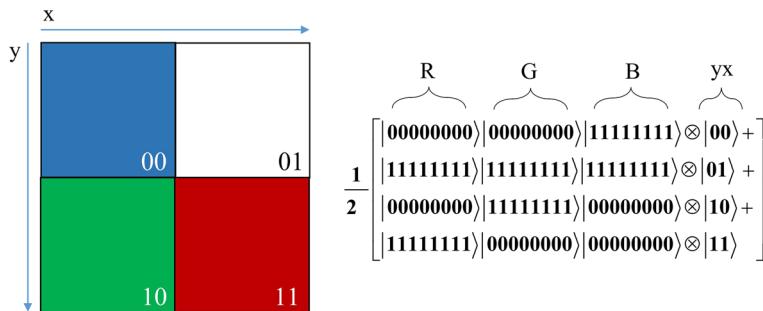


Fig. 1 A simple example of 2×2 RGB image and its NCQI quantum representation

Hence, there are three parts, namely the color information C_{RGB_i} , the vertical position y and the horizontal position x to depict one pixel. To exemplify the mentioned explanation, a simple 2×2 RGB image and its NCQI representation are illustrated in Fig. 1.

3 A dual quantum image scrambling method

In this section, inspired by the proposed method implemented on the digital images in Ref. [36], a new dual quantum scrambling method for RGB NCQI images is suggested.

To begin with, deem that P is a $2^n \times 2^n$ RGB image, the NCQI representation of which can be shown as below:

$$|P\rangle = \frac{1}{2^n} \sum_{i=1}^{2^{2n}} |P_{RGB_i}\rangle \otimes |i\rangle. \quad (4)$$

The proposed method is divided in two sub-algorithms which we will describe in two separate sections illustratively. It is worth mentioning that the first algorithm manipulates qubits of the RGB channels by XOR and XNOR operations, which is called bit-plane scrambler. Afterward, disordering pixels of the image is implemented by the second algorithm called pixel-plane scrambler. The scrambled output of the first algorithm is taken into account as the input of the second algorithm. The procedure of the mentioned operations is demonstrated in Fig. 2.

3.1 The first algorithm: bit-plane scrambler

As we know, in the RGB images, three channels are devoted to the color information in the sense that each channel includes eight bits (in an NCQI colored image, we illustratively have eight qubits for each channel). In this section, we explain our first method just by considering one channel and then all channels are engaged by the algorithm. Figure 3 indicates the outline of our bit-plane scrambler.

| a | b | a Xor b | a | b | a Xnor b |
|----------|----------|----------------|----------|----------|-----------------|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |

(a)

(b)

Fig. 2 The XOR and XNOR operations. **a** XOR and **b** XNOR

A brief look at the outline explicitly reveals that our bit-plane scrambler has two steps, showing different ways of changing color information. With respect to one channel, it has eight qubits and the most valuable qubits including 8, 7, 6 and 5 are accommodated by the results of the XOR and XNOR operations calculated by the first step, which is depicted in Fig. 3a. The second step displayed in Fig. 3b modifies the least valuable qubits including 1, 2, 3 and 4 by implementing XOR and XNOR operations on the existent qubits according to the outline. The solid and dash arrows in the outline refer to the XOR and XNOR operations in turn.

In order to employ the bit-plane scrambler, Step-1 and Step-2 are applied to the channel, respectively. Not only does our bit-plane scrambler involves all three channels in a pixel, but also it is of crucial importance that all of the pixels in the original image become engaged in the procedure completely. To clarify perceptively, in the following part, the mechanism of the bit-plane scrambler is shown in detail.

Step 1

- Qubit 8 \leftarrow qubit 8 *XOR* qubit 1
- Qubit 7 \leftarrow qubit 7 *XNOR* qubit 2
- Qubit 6 \leftarrow qubit 6 *XOR* qubit 3
- Qubit 5 \leftarrow qubit 5 *XNOR* qubit 4

Step 2

- Qubit 4 \leftarrow qubit 4 *XNOR* qubit 8
- Qubit 3 \leftarrow qubit 3 *XOR* qubit 7
- Qubit 2 \leftarrow qubit 2 *XNOR* qubit 6
- Qubit 1 \leftarrow qubit 1 *XOR* qubit 5

To design the quantum circuit of the bit-plane scrambler, we utilize the CNOT gates in order to implement XOR and XNOR operations on the appropriate qubits. The pertained quantum circuit drawn for this section is depicted in Fig. 4.

The derivation of the bit-plane scrambler can be written by the transformation operation \mathcal{Q} that is applied to the original image in order to be scrambled.

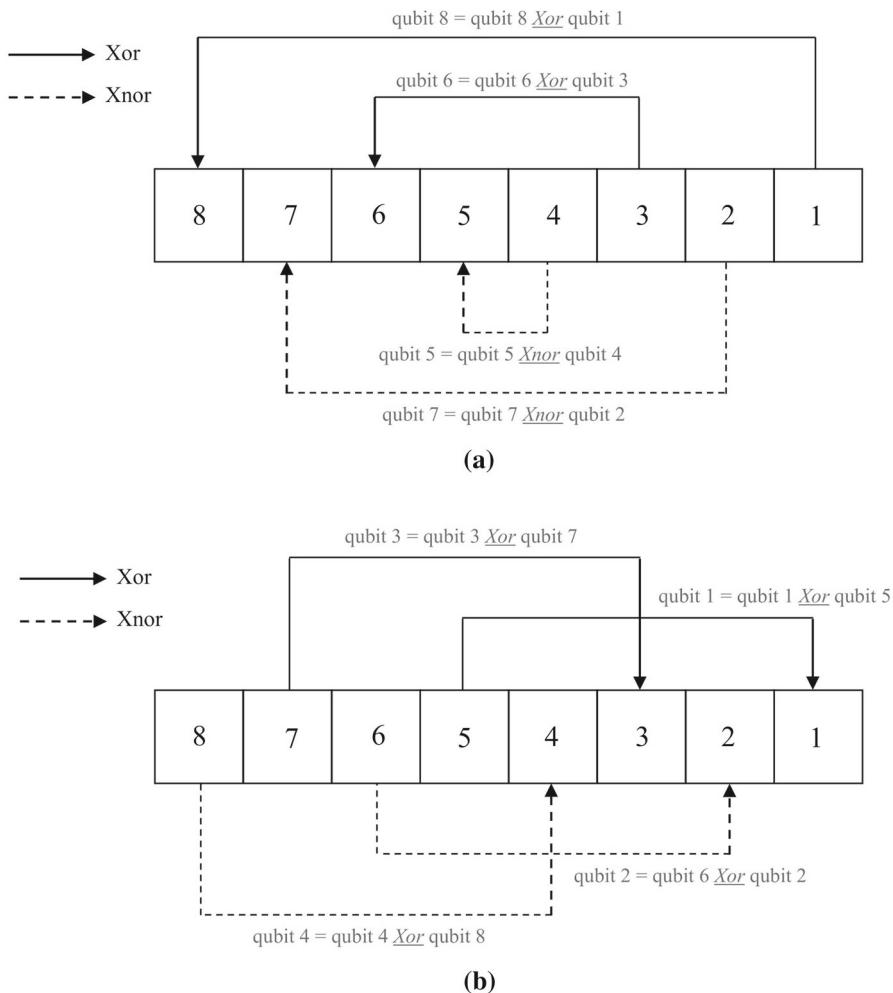


Fig. 3 The outline of the bit-plane scrambler. **a** Step one of the bit-plane scrambler and **b** step two of the bit-plane scrambler

$$\Omega_i = I \otimes \Omega \otimes |i\rangle\langle i| + I \otimes \left(\sum_{j=1, j \neq i}^{2^n} |j\rangle\langle j| \right), \quad (5)$$

where

$$\Omega = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

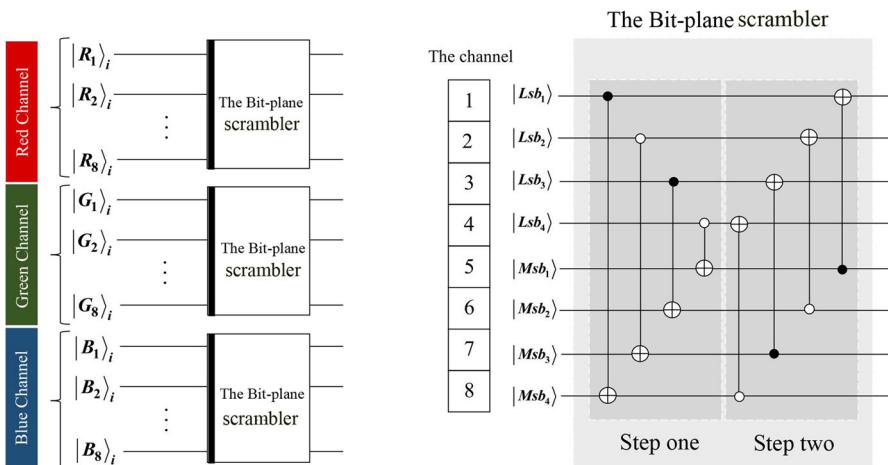


Fig. 4 The designed quantum circuit for executing the bit-plane scrambler

\mathcal{Q} is a CNOT gate required to be applied to the original image $|P\rangle$.

$$\begin{aligned}
 \mathcal{Q}_i |P\rangle &= \left(I \otimes \mathcal{Q} \otimes |i\rangle\langle i| + I \otimes \left(\sum_{j=1, j \neq i}^{2^n} |j\rangle\langle j| \right) \right) \left(\frac{1}{2^n} \sum_{i=1}^{2^n} |P_{RGBi}\rangle|i\rangle \right) \\
 &= \frac{1}{2^n} \left(I \otimes \mathcal{Q} \otimes |i\rangle\langle i| + I \otimes \left(\sum_{j=1, j \neq i}^{2^n} |j\rangle\langle j| \right) \right) (|P_{RGBi}\rangle|i\rangle) \\
 &\quad + \frac{1}{2^n} \left(I \otimes \mathcal{Q} \otimes |i\rangle\langle i| + I \otimes \left(\sum_{j=1, j \neq i}^{2^n} |j\rangle\langle j| \right) \right) \left(\sum_{j=1, j \neq i}^{2^n} |P_{RGBj}\rangle|j\rangle \right) \\
 &= \frac{1}{2^n} (I \otimes \mathcal{Q} \otimes |i\rangle\langle i|(|P_{RGBi}\rangle|i\rangle)) \\
 &\quad + \frac{1}{2^n} \left(I \otimes \left(\sum_{j=1, j \neq i}^{2^n} |j\rangle\langle j| \right) \right) \left(\sum_{j=0, j \neq i}^{2^n} |P_{RGBj}\rangle|j\rangle \right) \\
 &= \frac{1}{2^n} \mathcal{Q} |P_{RGBi}\rangle|i\rangle + \frac{1}{2^n} \sum_{j=1, j \neq i}^{2^n} |P_{RGBj}\rangle|j\rangle \\
 &= \frac{1}{2^n} \mathcal{Q} (|R_i\rangle|G_i\rangle|B_i\rangle)|i\rangle + \frac{1}{2^n} \sum_{j=1, j \neq i}^{2^n} (|R_j\rangle|G_j\rangle|B_j\rangle)|j\rangle \\
 &= \frac{1}{2^n} (|\bar{R}_i\rangle|\bar{G}\rangle|\bar{B}_i\rangle)|i\rangle + \frac{1}{2^n} \sum_{j=1, j \neq i}^{2^n} (|R_j\rangle|G_j\rangle|B_j\rangle)|j\rangle.
 \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2^n} (|P'_{\text{RGB}i}\rangle) |i\rangle + \frac{1}{2^n} \sum_{j=1, j \neq i}^{2^{2n}} (|P'_{\text{RGB}j}\rangle) |j\rangle. \\
&= \frac{1}{2^n} \sum_{i=1}^{2^{2n}} |P'_{\text{RGB}i}\rangle |i\rangle \\
&= |P'\rangle_1
\end{aligned} \tag{6}$$

where according to our bit-plane scrambler:

Red channel:

- If $|R_i\rangle = |r_i^8 r_i^7 r_i^6 r_i^5 r_i^4 r_i^3 r_i^2 1\rangle$ then $|\bar{R}_i\rangle = |(\sim r_i^8) r_i^7 r_i^6 r_i^5 r_i^4 r_i^3 r_i^2 r_i^1\rangle$
- If $|R_i\rangle = |r_i^8 r_i^7 r_i^6 r_i^5 r_i^4 r_i^3 0 r_i^1\rangle$ then $|\bar{R}_i\rangle = |r_i^8 (\sim r_i^7) r_i^6 r_i^5 r_i^4 r_i^3 r_i^2 r_i^1\rangle$
- If $|R_i\rangle = |r_i^8 r_i^7 r_i^6 r_i^5 r_i^4 1 r_i^2 r_i^1\rangle$ then $|\bar{R}_i\rangle = |r_i^8 r_i^7 (\sim r_i^6) r_i^5 r_i^4 r_i^3 r_i^2 r_i^1\rangle$
- If $|R_i\rangle = |r_i^8 r_i^7 r_i^6 r_i^5 0 r_i^3 r_i^2 r_i^1\rangle$ then $|\bar{R}_i\rangle = |r_i^8 r_i^7 r_i^6 (\sim r_i^5) r_i^4 r_i^3 r_i^2 r_i^1\rrangle$
- If $|R_i\rangle = |0 r_i^7 r_i^6 r_i^5 r_i^4 r_i^3 r_i^2 r_i^1\rangle$ then $|\bar{R}_i\rangle = |r_i^8 r_i^7 r_i^6 r_i^5 (\sim r_i^4) r_i^3 r_i^2 r_i^1\rrangle$
- If $|R_i\rangle = |r_i^8 1 r_i^6 r_i^5 r_i^4 r_i^3 r_i^2 r_i^1\rangle$ then $|\bar{R}_i\rangle = |r_i^8 r_i^7 r_i^6 r_i^5 r_i^4 (\sim r_i^3) r_i^2 r_i^1\rrangle$
- If $|R_i\rangle = |r_i^8 r_i^7 0 r_i^5 r_i^4 r_i^3 r_i^2 r_i^1\rangle$ then $|\bar{R}_i\rangle = |r_i^8 r_i^7 r_i^6 r_i^5 r_i^4 r_i^3 (\sim r_i^2) r_i^1\rrangle$
- If $|R_i\rangle = |r_i^8 r_i^7 r_i^6 1 r_i^4 r_i^3 r_i^2 r_i^1\rangle$ then $|\bar{R}_i\rangle = |r_i^8 r_i^7 r_i^6 r_i^5 r_i^4 r_i^3 r_i^2 (\sim r_i^1)\rangle$

Green channel:

- If $|G_i\rangle = |g_i^8 g_i^7 g_i^6 g_i^5 g_i^4 g_i^3 g_i^2 1\rangle$ then $|\bar{G}_i\rangle = |(\sim g_i^8) g_i^7 g_i^6 g_i^5 g_i^4 g_i^3 g_i^2 g_i^1\rrangle$
- If $|G_i\rangle = |g_i^8 g_i^7 g_i^6 g_i^5 g_i^4 g_i^3 0 g_i^1\rangle$ then $|\bar{G}_i\rangle = |g_i^8 (\sim g_i^7) g_i^6 g_i^5 g_i^4 g_i^3 g_i^2 g_i^1\rrangle$
- If $|G_i\rangle = |g_i^8 g_i^7 g_i^6 g_i^5 g_i^4 1 g_i^2 g_i^1\rangle$ then $|\bar{G}_i\rangle = |g_i^8 g_i^7 (\sim g_i^6) g_i^6 g_i^5 g_i^4 g_i^3 g_i^2 g_i^1\rrangle$
- If $|G_i\rangle = |g_i^8 g_i^7 g_i^6 g_i^5 0 g_i^3 g_i^2 g_i^1\rangle$ then $|\bar{G}_i\rangle = |g_i^8 g_i^7 g_i^6 (\sim g_i^5) g_i^4 g_i^3 g_i^2 g_i^1\rrangle$
- If $|G_i\rangle = |0 g_i^7 g_i^6 g_i^5 g_i^4 g_i^3 g_i^2 g_i^1\rangle$ then $|\bar{G}_i\rangle = |g_i^8 g_i^7 g_i^6 (\sim g_i^4) g_i^3 g_i^2 g_i^1\rrangle$
- If $|G_i\rangle = |g_i^8 1 g_i^6 g_i^5 g_i^4 g_i^3 g_i^2 g_i^1\rangle$ then $|\bar{G}_i\rangle = |g_i^8 g_i^7 g_i^6 g_i^5 g_i^4 (\sim g_i^3) g_i^2 g_i^1\rrangle$
- If $|G_i\rangle = |g_i^8 g_i^7 0 g_i^5 g_i^4 g_i^3 g_i^2 g_i^1\rangle$ then $|\bar{G}_i\rangle = |g_i^8 g_i^7 g_i^6 g_i^5 g_i^4 g_i^3 (\sim g_i^2) g_i^1\rrangle$
- If $|G_i\rangle = |g_i^8 g_i^7 g_i^6 1 g_i^4 g_i^3 g_i^2 g_i^1\rangle$ then $|\bar{G}_i\rangle = |g_i^8 g_i^7 g_i^6 g_i^5 g_i^4 g_i^3 (\sim g_i^1)\rangle$

Blue channel:

- If $|B_i\rangle = |b_i^8 b_i^7 b_i^6 b_i^5 b_i^4 b_i^3 b_i^2 1\rangle$ then $|\bar{B}_i\rangle = |(\sim b_i^8) b_i^7 b_i^6 b_i^5 b_i^4 b_i^3 b_i^2 b_i^1\rrangle$
- If $|B_i\rangle = |b_i^8 b_i^7 b_i^6 b_i^5 b_i^4 b_i^3 b_i^2 0 b_i^1\rangle$ then $|\bar{B}_i\rangle = |b_i^8 (\sim b_i^7) b_i^6 b_i^5 b_i^4 b_i^3 b_i^2 b_i^1\rrangle$
- If $|B_i\rangle = |b_i^8 b_i^7 b_i^6 b_i^5 b_i^4 1 b_i^2 b_i^1\rangle$ then $|\bar{B}_i\rangle = |b_i^8 b_i^7 (\sim b_i^6) b_i^5 b_i^4 b_i^3 b_i^2 b_i^1\rrangle$
- If $|B_i\rangle = |b_i^8 b_i^7 b_i^6 b_i^5 0 b_i^3 b_i^2 b_i^1\rangle$ then $|\bar{B}_i\rangle = |b_i^8 b_i^7 b_i^6 (\sim b_i^5) b_i^4 b_i^3 b_i^2 b_i^1\rrangle$
- If $|B_i\rangle = |0 b_i^7 b_i^6 b_i^5 b_i^4 b_i^3 b_i^2 b_i^1\rangle$ then $|\bar{B}_i\rangle = |b_i^8 b_i^7 b_i^6 b_i^5 (\sim b_i^4) b_i^3 b_i^2 b_i^1\rrangle$
- If $|B_i\rangle = |b_i^8 1 b_i^6 b_i^5 b_i^4 b_i^3 b_i^2 b_i^1\rangle$ then $|\bar{B}_i\rangle = |b_i^8 b_i^7 b_i^6 b_i^5 b_i^4 (\sim b_i^3) b_i^2 b_i^1\rrangle$
- If $|B_i\rangle = |b_i^8 b_i^7 0 b_i^5 b_i^4 b_i^3 b_i^2 b_i^1\rangle$ then $|\bar{B}_i\rangle = |b_i^8 b_i^7 b_i^6 b_i^5 b_i^4 b_i^3 (\sim b_i^2) b_i^1\rrangle$
- If $|B_i\rangle = |b_i^8 b_i^7 b_i^6 1 b_i^4 b_i^3 b_i^2 b_i^1\rangle$ then $|\bar{B}_i\rangle = |b_i^8 b_i^7 b_i^6 b_i^5 b_i^4 b_i^3 b_i^2 (\sim b_i^1)\rangle$

By applying the operation $\prod_{i=1}^{2^{2n}} \Omega_i$ to the original image $|P\rangle$, the scrambled image $|P'\rangle$ is produced. The output image generated in this section is given to the next part of our proposed approach called the pixel-plane scrambler.

3.2 The second algorithm: pixel-plane scrambler

In this section, we describe our pixel-plane scrambler as the second algorithm in order to scramble the NCQI image. This algorithm is implemented on T which is the output of the bit-plane scrambler. Likewise, the pixel-plane scrambler comprises two steps, dividing T in two parts and dedicating Step-1 and Step-2 to the first and second parts of T in turn. The outline of the pixel-plane scrambler is depicted in Fig. 5.

In the outline of the pixel-plane scrambler depicted in Fig. 5, m is 2^{2n} and it is just designed for one channel. The solid arrows show the XOR operations and the XNOR operations are illustrated by dash arrows. In this scrambler, pixels of the image T are scrambled by using qubits of the other pixels. In both steps, the XOR and XNOR operations are applied to even and odd qubits of the pixels, respectively. To scramble the first half of the image, the Step-1 is considered and the Step-2 is dedicated to the second half of the pixels in the image T , which can be shown as follows.

– First step

- Even plane: 8th, 6th, 4th and 2nd qubits:
 - (The even qubit in pixel 1) \leftarrow (The even qubit in pixel m) XOR (The even qubit in pixel 1)
 - (The even qubit in pixel 2) \leftarrow (The even qubit in pixel $m - 1$) XOR (The even qubit in pixel 2)
 - (The even qubit in pixel 3) \leftarrow (The even qubit in pixel $m - 2$) XOR (The even qubit in pixel 3)
 - .
 - .
 - .
 - (The even qubit in pixel $m/2$) \leftarrow (The even qubit in pixel $(m/2) + 1$) XOR (The even qubit in pixel $m/2$)
- Odd plane: 7th, 5th, 3rd and 1st qubits:
 - (The odd qubit in pixel 1) \leftarrow (The odd qubit in pixel m) XNOR (The odd qubit in pixel 1)
 - (The odd qubit in pixel 2) \leftarrow (The odd qubit in pixel $m - 1$) XNOR (The odd qubit in pixel 2)
 - (The odd qubit in pixel 3) \leftarrow (The odd qubit in pixel $m - 2$) XNOR (The odd qubit in pixel 3)
 - .
 - .
 - .
 - (The odd qubit in pixel $m/2$) \leftarrow (The odd qubit in pixel $(m/2) + 1$) XNOR (The odd qubit in pixel $m/2$)

– Second step

- Even plane: 8th, 6th, 4th and 2nd qubits:
 - (The even qubit in pixel $(m/2)+1$) \leftarrow (The even qubit in pixel $(m/2) + 1$) XOR (The even qubit in pixel 1)
 - (The even qubit in pixel $(m/2)+2$) \leftarrow (The even qubit in pixel $(m/2) + 2$) XOR (The even qubit in pixel 2)

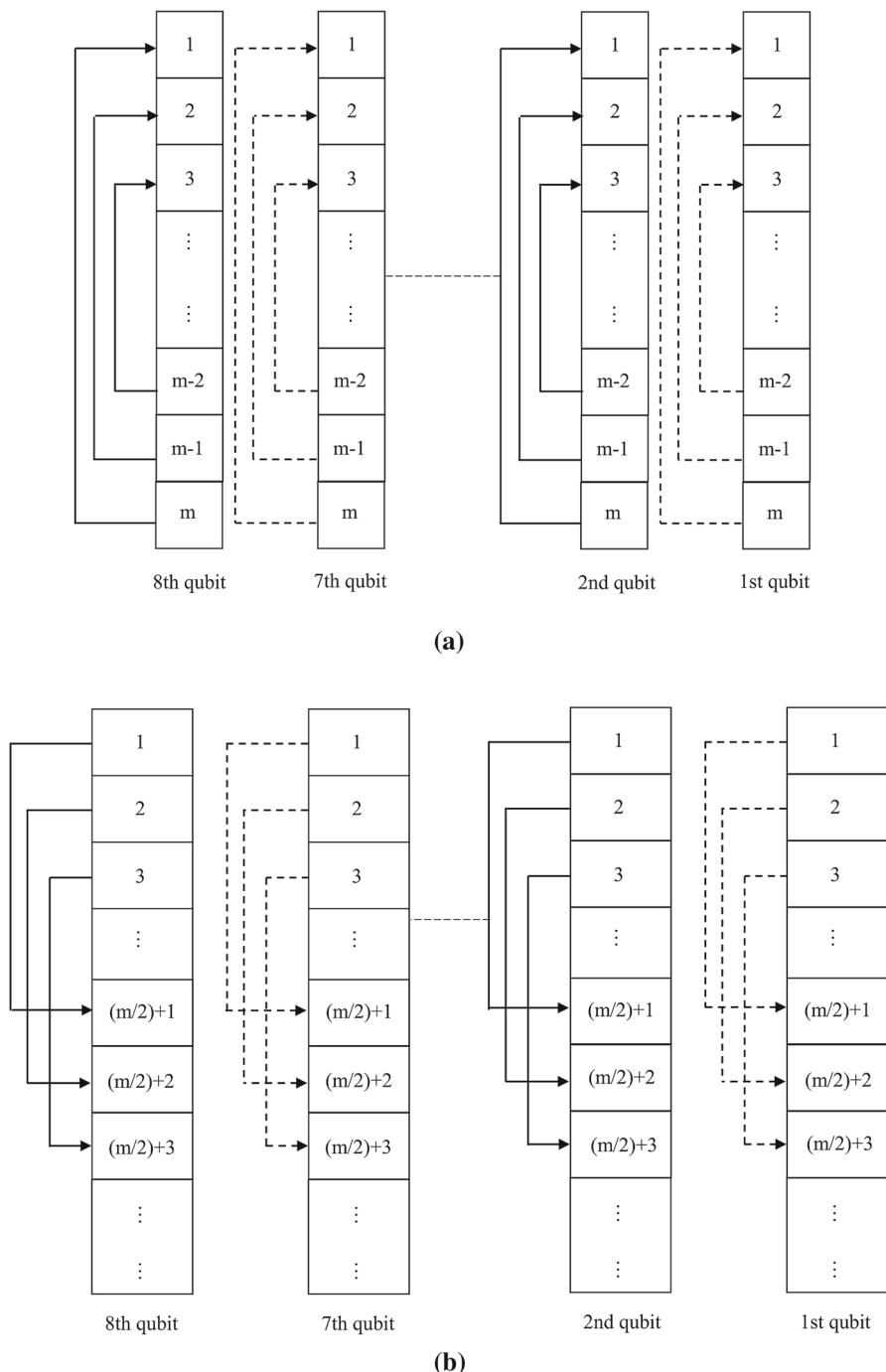


Fig. 5 The outline of the pixel-plane scrambler. **a** Step one of the pixel-plane scrambler and **b** step two of the pixel-plane scrambler

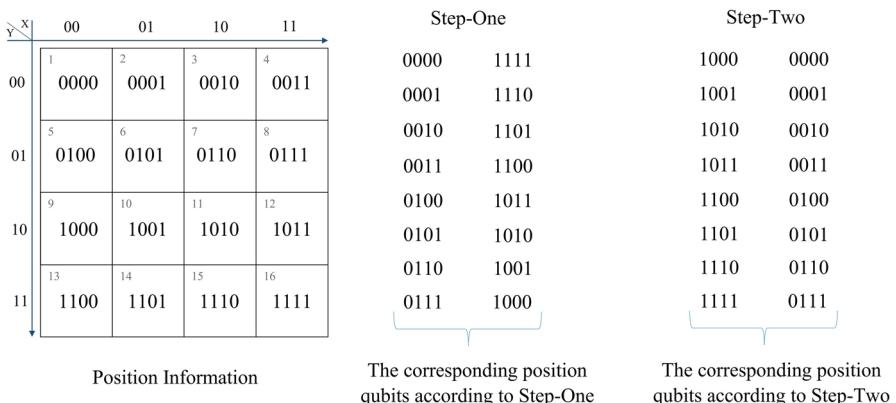


Fig. 6 The illustration of how can find a logic link between the two position-qubits according to each step.

- (The even qubit in pixel $(m/2)+3 \leftarrow$ (The even qubit in pixel $(m/2) + 3$) XOR (The even qubit in pixel 3))
·
·
• (The even qubit in pixel $m \leftarrow$ (The even qubit in pixel (m)) XOR (The even qubit in pixel $m/2$))
– Odd plane: 7th, 5th, 3rd and 1st qubits:
 - (The odd qubit in pixel $(m/2)+1 \leftarrow$ (The odd qubit in pixel $(m/2) + 1$) XNOR (The odd qubit in pixel 1))
 - (The odd qubit in pixel $(m/2)+2 \leftarrow$ (The odd qubit in pixel $(m/2) + 2$) XNOR (The odd qubit in pixel 2))
 - (The odd qubit in pixel $(m/2)+3 \leftarrow$ (The odd qubit in pixel $(m/2) + 3$) XNOR (The odd qubit in pixel 3))
·
• (The odd qubit in pixel $m \leftarrow$ (The odd qubit in pixel (m)) XNOR (The odd qubit in pixel $m/2$))

Initially, to access the related position information in an NCQI image, according to the two steps of the algorithm, a suitable quantum circuit should be designed. As a matter of fact, it is of vital importance to know how position information can be determined without affecting other qubits. Figure 6 illustrates how a clear connection can be found between the two pertinent location qubits according to each step.

Looking at details, what stands out from Fig. 6 is that in Step-1, the two corresponding position-qubits are exactly inverse, which would enable each of them to turn into the other just by applying a CNOT gate. Turning to the second step, what is obvious in the comparison of the mentioned pairs is that most significant qubit (MSQb) of position-qubits is the only difference. Consequently, what is needed is just passing the first qubit through a CNOT gate. Take, for example, 0000 which is the first position in an NCQI image. To find its corresponding position according to Step-1, inverting of

the qubits should be regarded, which would be 1111. Thus, the first position and the last one in Step-1 would be the related pair. Moreover, for finding the related position-qubits for 0000 according to Step-2, its first qubit needs to be inverted, which would be 1000. So, as can be seen, there is an explicit link between the patterns in two steps.

Before designing the related quantum circuit, in this step we need define $|T'\rangle$ which is the same image as $|T\rangle$, meaning that both images have the same size and they include same information. According to NCQI, this image needs $2n + 3q$ qubits, which here q equals 8. Initially, this image can be defined as follows.

$$|T'\rangle = \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} |T'_{\text{RGB}_i}\rangle \otimes |i\rangle, \quad (7)$$

$$|T'_{\text{RGB}_i}\rangle = |R'_i\rangle |G'_i\rangle |B'_i\rangle, \quad (8)$$

$$\begin{aligned} |R'_i\rangle &= |r'^{q-1}_i r'^{q-2}_i \dots r'^1_i r'^0_i\rangle, \\ |G'_i\rangle &= |g'^{q-1}_i g'^{q-2}_i \dots g'^1_i g'^0_i\rangle, \\ |B'_i\rangle &= |b'^{q-1}_i b'^{q-2}_i \dots b'^1_i b'^0_i\rangle, \\ r'^k_i, g'^k_i, b'^k_i &= 0 \end{aligned}$$

and also $|i'\rangle$ is shown as below.

$$|i'\rangle = |y'\rangle |x'\rangle = |y'_{n-1}, y'_{n-2} \dots y'_0\rangle |x'_{n-1}, x'_{n-2} \dots x'_0\rangle, y'_j x'_j \in \{0, 1\}, \quad (9)$$

The required quantum circuit for making $|T'\rangle$ from $|T\rangle$ and performing the pixel-plane scrambler can be illustrated in Figs. 7 and 8 in turn.

The derivation of the pixel-plane scrambler can be construed as follows.

As mentioned previously, this step of the proposed method itself has two steps, manipulating half of the image. Firstly, a transformation operation named $|\Gamma^1\rangle$ is defined, which is applied to the image $|T\rangle$ generated from the bit-plane scrambler.

$$\Gamma_i^1 = I \otimes \Gamma^1 \otimes |i\rangle\langle i| + I \otimes \left(\sum_{j=1, j \neq i}^{2^{2n}} |j\rangle\langle j| \right), \quad (10)$$

where

$$\Gamma^1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Γ^1 is a CNOT gate and it is required to be applied to the image $|T\rangle$.

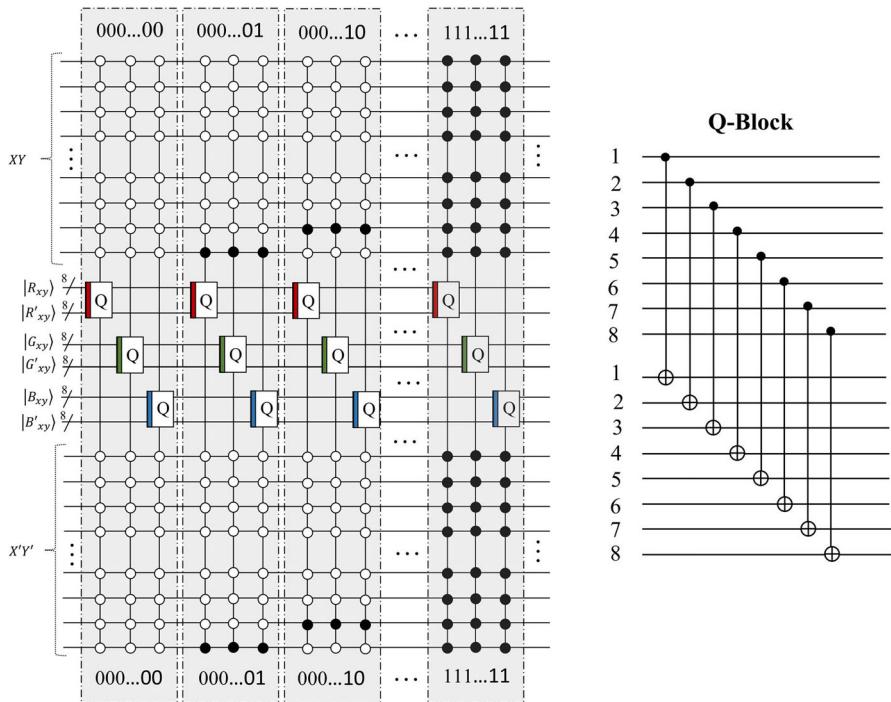


Fig. 7 The designed quantum circuit for making T'

$$\begin{aligned}
 \Gamma_i^1 |T\rangle &= \left(I \otimes \Gamma^1 \otimes |i\rangle\langle i| + I \otimes \left(\sum_{j=1, j \neq i}^{2^n} |j\rangle\langle j| \right) \right) \left(\frac{1}{2^n} \sum_{i=1}^{2^n} |T_{RGBi}\rangle|i\rangle \right) \\
 &= \frac{1}{2^n} \left(I \otimes \Gamma^1 \otimes |i\rangle\langle i| + I \otimes \left(\sum_{j=1, j \neq i}^{2^n} |j\rangle\langle j| \right) \right) (|T_{RGBi}\rangle|i\rangle) \\
 &\quad + \frac{1}{2^n} \left(I \otimes \Gamma^1 \otimes |i\rangle\langle i| + I \otimes \left(\sum_{j=1, j \neq i}^{2^n} |j\rangle\langle j| \right) \right) \left(\sum_{j=1, j \neq i}^{2^n} |T_{RGBj}\rangle|j\rangle \right) \\
 &= \frac{1}{2^n} (I \otimes \Gamma^1 \otimes |i\rangle\langle i| (|T_{RGBi}\rangle|i\rangle)) \\
 &\quad + \frac{1}{2^n} \left(I \otimes \left(\sum_{j=1, j \neq i}^{2^n} |j\rangle\langle j| \right) \right) \left(\sum_{j=0, j \neq i}^{2^n} |T_{RGBj}\rangle|j\rangle \right) \\
 &= \frac{1}{2^n} \Gamma^1 |T_{RGBi}\rangle|i\rangle + \frac{1}{2^n} \sum_{j=1, j \neq i}^{2^n} |T_{RGBj}\rangle|j\rangle
 \end{aligned}$$

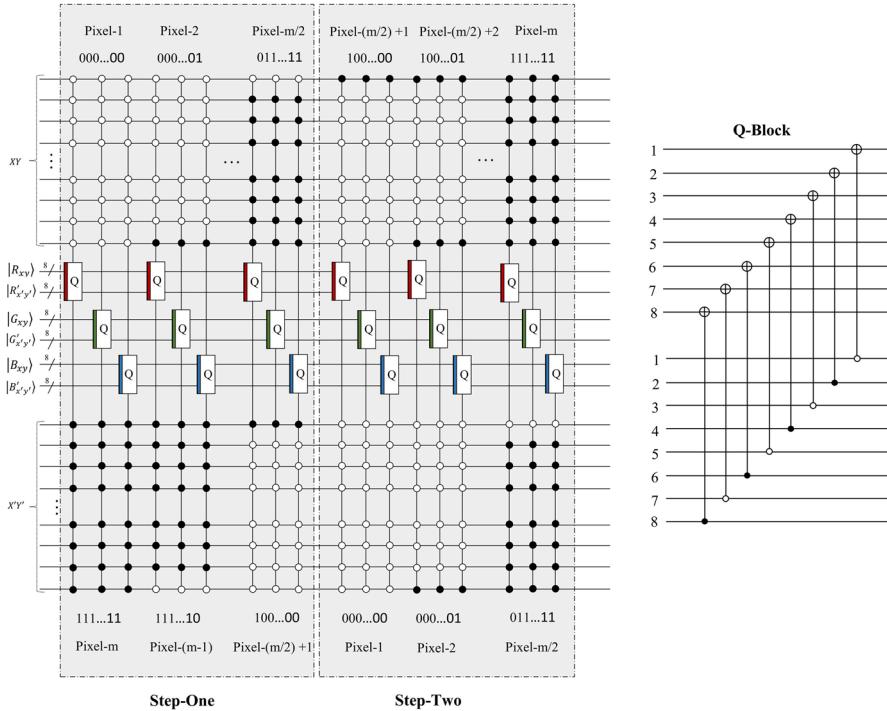


Fig. 8 The designed quantum circuit for the pixel-plane scrambler

$$\begin{aligned}
 &= \frac{1}{2^n} \Gamma^1(|R_i\rangle|G_i\rangle|B_i\rangle)|i\rangle + \frac{1}{2^n} \sum_{j=1, j \neq i}^{2^{2n}} (|R_j\rangle|G_j\rangle|B_j\rangle)|j\rangle \\
 &= \frac{1}{2^n} (|\bar{R}_i\rangle|\bar{G}\rangle|\bar{B}_i\rangle)|i\rangle + \frac{1}{2^n} \sum_{j=1, j \neq i}^{2^{2n}} (|R_j\rangle|G_j\rangle|B_j\rangle)|j\rangle. \\
 &= \frac{1}{2^n} (|\bar{T}_{\text{RGB}i}\rangle)|i\rangle + \frac{1}{2^n} \sum_{j=1, j \neq i}^{2^{2n}} (|T_{\text{RGB}j}\rangle)|j\rangle. \\
 &= \frac{1}{2^n} \sum_{i=1}^{2^{2n}} |\bar{T}_{\text{RGB}i}\rangle|i\rangle \\
 &= |\bar{T}\rangle_1
 \end{aligned} \tag{11}$$

where according to the pixel-plane scrambler:

Red channel

- If $|R_i\rangle = |r_i^7 r_i^6 r_i^5 r_i^4 r_i^3 r_i^2 r_i^1\rangle$ then $|\bar{R}_k\rangle = |\sim(r_k^8 r_k^7 r_k^6 r_k^5 r_k^4 r_k^3 r_k^2 r_k^1)\rangle$ and $|\bar{R}_i\rangle$ is unchanged; $m = 2^{2n}$ and $(k = (m/2) + i) \leq m$

- If $|R_i\rangle = |r_i^8 r_i^7 1 r_i^5 r_i^4 r_i^3 r_i^2 r_i^1\rangle$ then $|\bar{R}_k\rangle = |r_k^8 r_k^7 \sim (r_k^6) r_k^5 r_k^4 r_k^3 r_k^2 r_k^1\rangle$ and $|\bar{R}_i\rangle$ is unchanged; $m = 2^{2n}$ and $(k = (m/2) + i) \leq m$
- If $|R_i\rangle = |r_i^8 r_i^7 r_i^6 r_i^5 1 r_i^3 r_i^2 r_i^1\rangle$ then $|\bar{R}_k\rangle = |r_k^8 r_k^7 r_k^6 r_k^5 \sim (r_k^4) r_k^3 r_k^2 r_k^1\rangle$ and $|\bar{R}_i\rangle$ is unchanged; $m = 2^{2n}$ and $(k = (m/2) + i) \leq m$
- If $|R_i\rangle = |r_i^8 r_i^7 r_i^6 r_i^5 r_i^4 r_i^3 1 r_i^1\rangle$ then $|\bar{R}_k\rangle = |r_k^8 r_k^7 r_k^6 r_k^5 r_k^4 r_k^3 \sim (r_k^2) r_k^1\rangle$ and $|\bar{R}_i\rangle$ is unchanged; $m = 2^{2n}$ and $(k = (m/2) + i) \leq m$

Green channel

- If $|G_i\rangle = |1 g_i^7 g_i^6 g_i^5 g_i^4 g_i^3 g_i^2 g_i^1\rangle$ then $|\bar{G}_k\rangle = | \sim (g_k^8) g_k^7 g_k^6 g_k^5 g_k^4 g_k^3 g_k^2 g_k^1\rangle$ and $|\bar{G}_i\rangle$ is unchanged; $m = 2^{2n}$ and $(k = (m/2) + i) \leq m$
- If $|G_i\rangle = |g_i^8 g_i^7 g_i^5 g_i^4 g_i^3 g_i^2 g_i^1\rangle$ then $|\bar{G}_k\rangle = |g_k^8 g_k^7 \sim (g_k^6) g_k^5 g_k^4 g_k^3 g_k^2 g_k^1\rangle$ and $|\bar{G}_i\rangle$ is unchanged; $m = 2^{2n}$ and $(k = (m/2) + i) \leq m$
- If $|G_i\rangle = |g_i^8 g_i^7 g_i^6 g_i^5 1 g_i^3 g_i^2 g_i^1\rangle$ then $|\bar{G}_k\rangle = |g_k^8 g_k^7 g_k^6 g_k^5 \sim (g_k^4) g_k^3 g_k^2 g_k^1\rangle$ and $|\bar{G}_i\rangle$ is unchanged; $m = 2^{2n}$ and $(k = (m/2) + i) \leq m$
- If $|G_i\rangle = |g_i^8 g_i^7 g_i^6 g_i^5 g_i^4 g_i^3 1 g_i^1\rangle$ then $|\bar{G}_k\rangle = |g_k^8 g_k^7 g_k^6 g_k^5 g_k^4 g_k^3 \sim (g_k^2) g_k^1\rangle$ and $|\bar{G}_i\rangle$ is unchanged; $m = 2^{2n}$ and $(k = (m/2) + i) \leq m$

Blue channel

- If $|B_i\rangle = |1 b_i^7 b_i^6 b_i^5 b_i^4 b_i^3 b_i^2 b_i^1\rangle$ then $|\bar{B}_k\rangle = | \sim (b_k^8) b_k^7 b_k^6 b_k^5 b_k^4 b_k^3 b_k^2 b_k^1\rangle$ and $|\bar{B}_i\rangle$ is unchanged; $m = 2^{2n}$ and $(k = (m/2) + i) \leq m$
- If $|B_i\rangle = |b_i^8 b_i^7 b_i^5 b_i^4 b_i^3 b_i^2 b_i^1\rangle$ then $|\bar{B}_k\rangle = |b_k^8 b_k^7 \sim (b_k^6) b_k^5 b_k^4 b_k^3 b_k^2 b_k^1\rangle$ and $|\bar{B}_i\rangle$ is unchanged; $m = 2^{2n}$ and $(k = (m/2) + i) \leq m$
- If $|B_i\rangle = |b_i^8 b_i^7 b_i^6 b_i^5 1 b_i^3 b_i^2 b_i^1\rangle$ then $|\bar{B}_k\rangle = |b_k^8 b_k^7 b_k^6 b_k^5 \sim (b_k^4) b_k^3 b_k^2 b_k^1\rangle$ and $|\bar{B}_i\rangle$ is unchanged; $m = 2^{2n}$ and $(k = (m/2) + i) \leq m$
- If $|B_i\rangle = |b_i^8 b_i^7 b_i^6 b_i^5 b_i^4 b_i^3 1 b_i^1\rangle$ then $|\bar{B}_k\rangle = |b_k^8 b_k^7 b_k^6 b_k^5 b_k^4 b_k^3 \sim (b_k^2) b_k^1\rangle$ and $|\bar{B}_i\rangle$ is unchanged; $m = 2^{2n}$ and $(k = (m/2) + i) \leq m$

Take (256×256) -RGB image named Lena in Fig. 9a, for instance. It can be scrambled by our dual quantum scrambling method, which can be seen in Fig. 9b.

By inverting the designed quantum circuits, the in-scrambling procedure can be performed on the scrambled image so that the original image could be captured completely. Accordingly, for the aim of the in-scrambling procedure, to begin, the second step and the first step of the pixel-plane scrambler must be applied to the scrambled image, respectively. Having performed the in-pixel-plane scrambler, the output then is considered as the input of the second step of the bit-plane scrambler. To conclude the process, the first step of the bit-plane scrambler is employed. The ultimate image would be precisely similar to the original image.

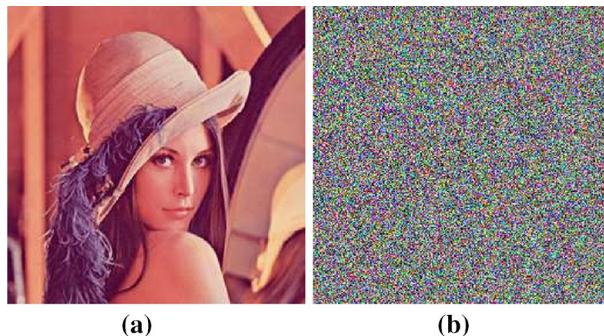


Fig. 9 The operation of our proposed method as an example, implementing on 256×256 Lena image

3.3 Circuit complexity

There is no doubt that for assessing a quantum network of circuits, it is of crucial importance that its complexity is acceptable. Thus, in this section, how our circuits' complexity could be calculated is shown as follows.

Generally, to extract the complexity, a rudimentary quantum gate is considered as a basic gate. Owing to the fact that only CNOT gates have been used in the investigated method, it will be our basic unit. With respect to the bit-plane scrambler shown in Fig. 4, the number of CNOT gates is 8 gates per channel. So we have 3×8 CNOT gates for each pixel, which means that by 3×8 CNOT gates, the bit-plane scrambler can be applied completely to all pixels due to the fact that in an NCQI image, pixels are in superposition states. As regards the second part of the algorithm, the pixel-plane scrambler, its complexity based on number of Toffoli gates can be easily calculated by considering the circuits depicted in Figs. 7 and 8. Due to the fact that size of the image is $2^n \times 2^n$ and NCQI method requires $n + n = 2n$ qubits for encoding the position information, the Toffoli gates include $2n$ controllers. According to Ref. [37], each m -qubit of the Toffoli needs at least $2m$ CNOT gates. Consequently, Fig. 7 showing how image T' is formed from T illustrates 3×8 $2n$ -Toffoli gates in order to make T' , meaning that it requires $(3 \times 8) \times (2 \times 2n) = (3 \times 8) \times 4n$ CNOT gates. Likewise, Fig. 8 displays 3×8 $2n$ -Toffoli gates to manipulate all pixels, which means that $(3 \times 8) \times 4n$ CNOT gates are required for this step. Thus, pixel-plane scrambler needs $2 \times ((3 \times 8) \times 4n)$.

To sum up, our quantum network complexity can be shown as below.

Circuit complexity = complexity of the bit-plane scrambler + complexity of the pixel-plane scrambler.

$$\begin{aligned} \text{Circuit complexity} &= (3 \times 8) + [2 \times ((3 \times 8) \times 4n)] \\ &= 192n + 24 \end{aligned} \quad (12)$$

This means that our dual quantum scrambling method could scramble a $2^2 \times 2^n$ -NCQI image by utilizing $192n + 24$ CNOT gates, when $q = 8$. Generally, regarding q , Eq. 12 can be shown.

$$\begin{aligned}\text{Circuit Complexity} &= (3 \times q) + [2 \times ((3 \times q) \times 4n)] \\ &= 24qn + 3q \text{ CNOT gates}\end{aligned}\quad (13)$$

4 Simulations and analysis

In order to evaluate our quantum scrambling method, in this section, the required simulation is performed, and some important criteria are analyzed. Due to the condition that the physical quantum hardware is not affordable for us to execute our method, our dual quantum scrambling technique is simulated with a classical computer Intel(R) Core (TM) i7-4500u CPU 2.40 GHz, 8.00 GB Ram equipped with the MATLAB R2015a environment. As the essential criteria, by which an image scrambling method can be assessed, *Histogram*, *Correlation Coefficient*, *Number of Pixels Change Ratio: NPCR* and *Entropy* are analyzed here.

To begin, Fig. 10 demonstrates the 256×256 images used in our simulation phase, depicting the corresponding scrambled images produced by our investigated method.

4.1 The histogram analysis

The image histogram diagram displays the frequency of the intensity values that occurs in an image. In fact, a histogram figure has two dimensions including the *x-axis* and the *y-axis*. The values of colors in the image is shown by *x-axis* and *y-axis* illustrating the number of pixels having the related color value. When an image is scrambled by changing the color value of pixels, the histogram diagram is also modified. The more flat the histogram diagram can be and the less peaks can be observed in the diagram, the more effective scrambling method we can have. By having a flat image histogram, more and more distribution of colors in a scrambled image can be found. As demonstrated in Fig. 11, our suggested method satisfies this criterion.

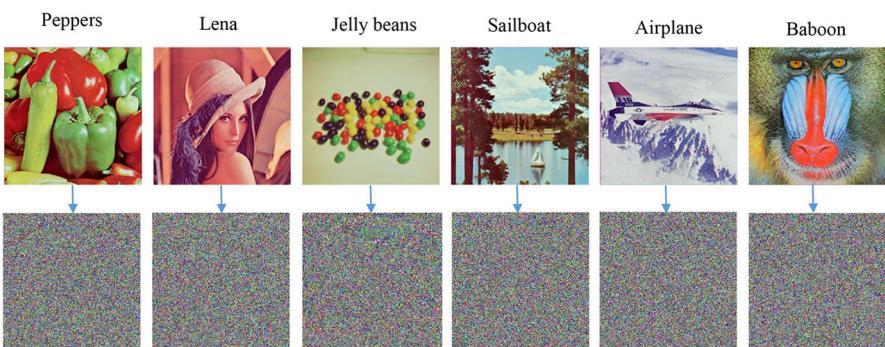


Fig. 10 The original images and the corresponding scrambled images

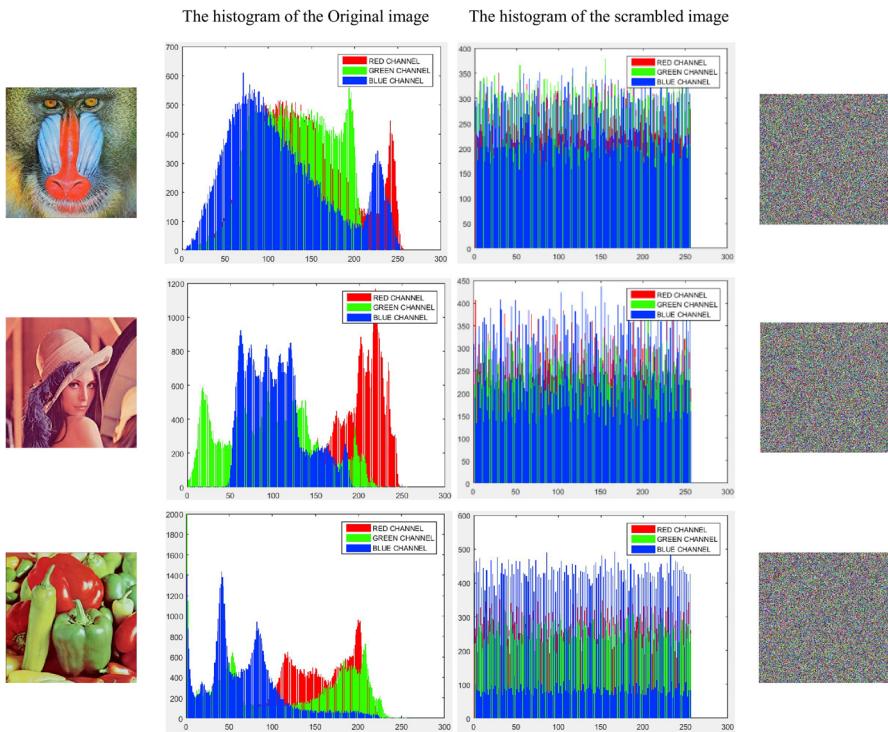


Fig. 11 Analyzing of the histogram diagrams resulted by the proposed technique

4.2 The correlation coefficient

The correlation between pixels in an image is calculated by the correlation coefficient. By scrambling an image, the correlation between pixels is destroyed and it is going to distribute. In image processing algorithms, an algorithm is considered as the best if the most devaluation in the correlation can be obtained. The Correlation Coefficient rates of our suggested method, in three coordinates including vertical (V), horizontal (H) and diagonal (D) Correlation Coefficient, are given in Table 1. These values have been calculated in the MATLAB environment by using its predefined functions.

Looking at table, our proposed algorithm has sufficient devaluation in the analyzing the correlation coefficient rates.

4.3 Number of pixels change ratio: NPCR

To analyze changing rate of pixels after scrambling procedure, NPCR values are calculated in our simulation. By comparing the color brightness of pixels in the original image and corresponding ones in the scrambled image, NPCR can be resulted. In image scrambling algorithms, the more NPCR values are close to 100%, the more efficiency can be gained. The NPCR formula is given as below.

Table 1 The correlation coefficient rates between pixels in the original and scrambled images

| Image name | The correlation coefficient (H) | The correlation coefficient (V) | The correlation coefficient (D) |
|-----------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Airplane—original | 0.9309 | 0.9278 | 0.8658 |
| Airplane—scrambled | 0.0127 | -0.0212 | -0.0131 |
| Baboon—original | 0.9125 | 0.8870 | 0.8577 |
| Baboon—scrambled | 0.0054 | -0.0169 | -0.0139 |
| Jelly beans—original | 0.9794 | 0.9801 | 0.9590 |
| Jelly beans—scrambled | 0.0070 | 0.0296 | 0.0013 |
| Lena—original | 0.9569 | 0.9781 | 0.9376 |
| Lena—scrambled | 0.0199 | -0.0077 | -0.0102 |
| Peppers—original | 0.9686 | 0.9762 | 0.9457 |
| Peppers—scrambled | 0.0251 | 0.0519 | -0.0131 |
| Sailboat—original | 0.9515 | 0.9511 | 0.9216 |
| Sailboat—scrambled | -0.0174 | 0.0335 | 0.0119 |

Table 2 The NPCR values between pixels in original images and scrambled images

| NPCR | Image |
|-------------|---------|
| Airplane | 99.6093 |
| Baboon | 99.6164 |
| Jelly beans | 99.6515 |
| Lena | 99.6093 |
| Peppers | 99.5936 |
| Sailboat | 99.6093 |

$$\text{NPCR} = \frac{\sum_i^{2^{2n}} D_i}{m} \times 100\% \\ D_i = \begin{cases} 1, & \text{If } OI_i \neq SI_i; \\ 0, & \text{If } OI_i = SI_i. \end{cases} \quad (14)$$

where m is the whole number of pixels that is 256×256 here. OI and SI are the original image and the scrambled image, respectively. By analyzing the NPCR values resulted in our simulation shown in Table 2, it can be conceived that our investigated approach satisfies this criterion.

4.4 Image entropy

The image entropy depicts the amount of information contained in an image. It can be elected as a measure of the detail provided by an image. If all pixels in an image have the same color intensity, this image will present the minimal entropy value. On the other hand, when each pixel of an image has a specific color intensity, the

Table 3 The calculated entropy values in our simulation

| Image | The entropy value (original image) | The entropy value (scrambled image) |
|-------------|---------------------------------------|--|
| Airplane | 6.6936 | 7.9125 |
| Baboon | 7.6922 | 7.9838 |
| Jelly beans | 6.8687 | 7.8678 |
| Lena | 7.7322 | 7.9742 |
| Peppers | 7.7249 | 7.9743 |
| Sailboat | 7.7526 | 7.9918 |

image will demonstrate the maximum entropy. An image scrambling procedure has a good efficiency when the scrambled image's entropy value is close to 8 [38]. The entropy values of our suggested method are indicated in Table 3 in which it can be seen obviously, the proposed method has acceptable entropy values. We have obtained these values by using predefined functions in the MATLAB environment.

5 Comparisons

This section is devoted to comparing our introduced method with the prior pertinent work proposed by Zhou et. al [35]. We simulate their work which encompasses two methods, in the MATLAB environment. The consequences of our comparisons are exemplified in the following subsections.

5.1 Circuit complexity

The investigated methods in Ref. [35] use an NEQR image with size $M \times N$, having just one channel to encode the gray-color information. If we apply their approaches on a $2^n \times 2^n$ NCQI image with three channels, the circuit complexity based on CNOT gate for their first method would be $3q$ and for the second one would be $2n + 3q$, while the complexity of our proposed method is $24qn + 3q$, meaning that their approach has better circuit complexity compared to what we suggest, requiring fewer CNOT gates to scramble the image. Despite this, the performance of their methods is significantly lower than our approach, and this new method is much reliable than theirs, as can be seen in the following sections.

5.2 Histogram comparison

As mentioned previously, after scrambling procedure, the more we can alter a histogram diagram to a flat diagram without peaks, the more efficient scrambling procedure we introduce. The histogram comparisons related to the Lena image are illustrated in Fig. 12. It is explicit that our proposed approach is efficient and applicable.

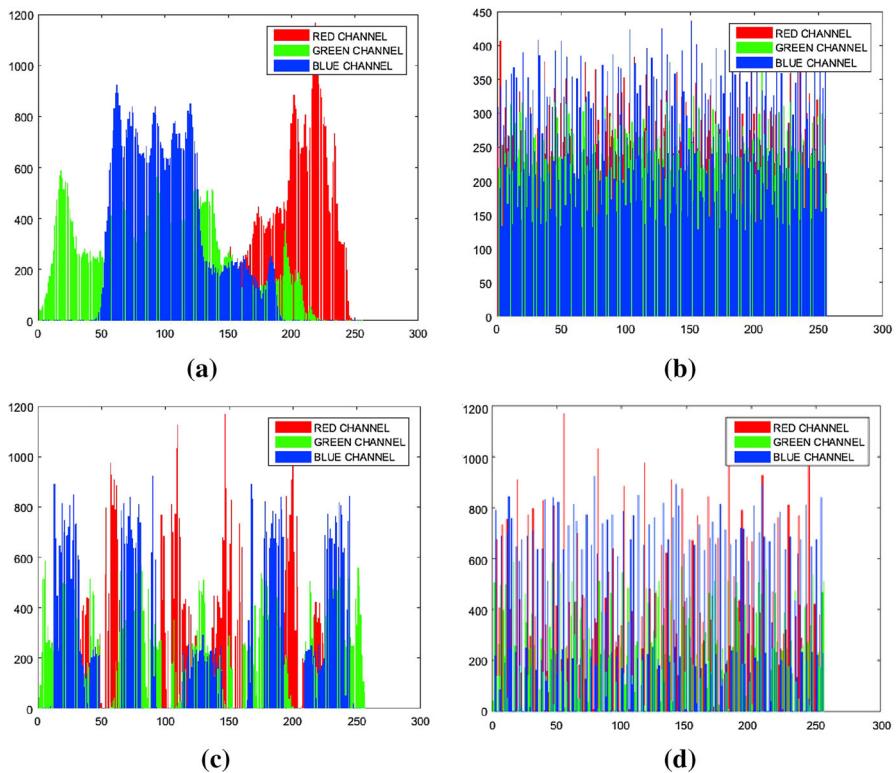


Fig. 12 Histogram comparison between our introduced method and the proposed methods in Ref. [35]
a Original image, **b** our proposed method, **c** the first introduced method in Ref. [35] and **d** the second introduced method in Ref. [35]

Table 4 The calculated entropy values for comparing our proposed method and the introduced methods in Ref. [35]

| | Original Lena | Our schema | First method of Ref. [35] | Second method of Ref. [35] |
|---------|---------------|------------|---------------------------|----------------------------|
| Entropy | 7.7322 | 7.9742 | 7.7322 | 7.7322 |

5.3 Entropy and correlation coefficient comparisons

In this section, we demonstrate our outcomes as two tables dedicated to entropy and correlation coefficient. As can be perceived from Tables 4 and 5, our suggested method has better values, for these criteria, than the introduced method in Ref. [35].

6 Conclusion

Image scrambling procedure is an approach by which a meaningful image is turned into a disordered meaningless image. The most noticeable application of this technique is

Table 5 The calculated correlation coefficient values for comparing our proposed method and the introduced methods in Ref. [35]

| | Horizontal | Vertical | Diagonal |
|----------------------------|------------|----------|----------|
| Original Lena | 0.9569 | 0.9781 | 0.9376 |
| Our schema | 0.0199 | -0.0077 | -0.0102 |
| First method of Ref. [35] | 0.0304 | 0.02933 | 0.0022 |
| Second method of Ref. [35] | 0.0418 | 0.0184 | -0.0004 |

when a steganography or watermarking algorithm is used. Due to the fact that various representations of quantum image have been suggested, quantum image scrambling can be employed as efficient as the similar digital image scrambling methods. In this contribution, a new dual quantum image scrambling technique in two levels has been investigated. Our image scrambling procedure has been done on the NCQI images effectively by applying two principle mechanisms including the bit-plane scrambler and the pixel-plane scrambler. In order to evaluate the performance of our investigated method, we have simulated the method in the MATLAB environment, where by calculating the NPCR, entropy, correlation coefficient and also capturing histogram graphs, the necessary requirements which all image scrambling procedures should meet have been confirmed. Moreover, it has been shown that our proposed technique outperforms a previous pertained quantum image scrambling method.

Acknowledgements The first author acknowledges the support of Kermanshah Branch, Young Researchers and Elite club, IRAN, and also would like to thank Besharat Rabiei, for her interest in this work.

References

1. Gupta, S., Goyal, A., Bhushan, B.: Information hiding using least significant bit steganography and cryptography. *Int. J. Educ. Comput. Sci.* **6**, 27–34 (2012)
2. Tsai, P., Hu, Y.U., Chang, C.C.: A color image watermarking scheme based on color quantization. *Signal Process.* **84**(1), 95–106 (2004)
3. Yan, F., Iliyasu, A.M., Venegas-Andraca, S.E.: A survey of quantum image representations. *Quantum Inf. Process.* **15**(1), 1–35 (2016)
4. Venegas-Andraca, S.E., Bose, S.: Storing, processing, and retrieving an image using quantum mechanics. In: Proceedings of SPIE Conference of Quantum Information and Computation, vol. 5105, pp. 134–147 (2003)
5. Latorre, J.: Image compression and entanglement. [arXiv:quant-ph/0510031](https://arxiv.org/abs/quant-ph/0510031) (2005)
6. Le, P., Dong, F., Hitora, K.: A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Inf. Process.* **10**(1), 63–84 (2011)
7. Sun, B., Iliyasu, A., Yan, F., Dong, F., Hitora, K.: An RGB multi-channel representation for images on quantum computers. *J. Adv. Comput. Intell. Intell. Inf.* **17**(3), 404–417 (2013)
8. Zhang, Y., Lu, K., Gao, Y., Wang, M.: NEQR: a novel enhanced quantum representation of digital images. *Quantum Inf. Process.* **12**(8), 2833–2860 (2013)
9. Zhang, Y., Lu, K., Gao, Y., Xu, K.: A novel quantum representation for log-polar images. *Quantum Inf. Process.* **12**(9), 3103–3126 (2013)
10. Yuan, S., Mao, X., Xue, Y., Chen, L., Xiong, Q., Compare, A.: SQR: a simple quantum representation of infrared images. *Quantum Inf. Process.* **13**(6), 1353–1379 (2014)
11. Abdolmaleky, M., et al.: Red-green-blue multi-channel quantum representation of digital images. *Optik* **128**, 121–132 (2017)

12. Sang, J.Z., Wang, S., Li, Q.: A novel quantum representation for color digital images. *Quantum Inf. Process.* **16**(2), 42 (2017)
13. Jiang, N., Dang, Y., Wang, J.: Quantum image matching. *Quantum Inf. Process.* **15**(0), 3543–3572 (2016)
14. Jiang, N., Dang, Y., Zhao, N.: Quantum image location. *Int. J. Theor. Phys.* **55**(10), 4501–4512 (2016)
15. Yuan, S., Mao, X., Zhou, J., Wang, X.: Quantum image filtering in the spatial domain. *Int. J. Theor. Phys.* **56**, 1–17 (2017)
16. Iliyasu, A., Le, P., Dong, F., Hitora, K.: Watermarking and authentication of quantum images based on restricted geometric transformations. *Inf. Sci.* **186**(1), 126–149 (2012)
17. Zhang, W., Gao, F., Liu, B., Jia, H.: A quantum watermark protocol. *Int. J. Theor. Phys.* **52**(2), 504–513 (2013)
18. Song, X., Wang, S., Liu, S., Abd El-Latif, A., Niu, X.: A dynamic watermarking scheme for quantum images using quantum wavelet transform. *Quantum Inf. Process.* **12**(2), 3689–3706 (2013)
19. Song, X., Wang, S., Abd El-Latif, A., Niu, X.: Dynamic watermarking scheme for quantum images based on hadamard transform. *Multimed. Syst.* **20**(4), 379–388 (2014)
20. Jiang, N., Wang, L.: A novel strategy for quantum image steganography based on Moiré pattern. *Int. J. Theor. Phys.* **54**(3), 1021–1032 (2015)
21. Wang, S., et al.: Least significant qubit (LSQb) information hiding algorithm for quantum image. *Measurement* **73**, 352–359 (2015)
22. Jiang, N., Zhao, N., Wang, L.: LSB based quantum image steganography algorithm. *Int. J. Theor. Phys.* **55**, 107–123 (2016)
23. Miyake, S., Nakamae, K.: A quantum watermarking scheme using simple and smallscale quantum circuits. *Quantum Inf. Process.* **15**, 1849–1864 (2016)
24. Sang, J., Wang, S., Li, Q.: Least significant qubit algorithm for quantum images. *Quantum Inf. Process.* **15**(11), 4441–4460 (2016)
25. Heidari, S., Naseri, M.: A novel LSB based quantum image watermarking. *Int. J. Theor. Phys.* **55**(10), 4205–4218 (2016)
26. Naseri, M., Heidari, S., et al.: A new secure quantum watermarking scheme. *Optik* **139**, 77–86 (2017)
27. Heidari, S., Gheibi, R., Houshmand, M., Nagata, K.: A robust blind quantum copyright protection method for colored images based on owner's signature. *Int. J. Theor. Phys.* **56**(8), 2562–2578 (2017)
28. Heidari, S., et al.: A new quantum watermarking based on quantum wavelet transforms. *Commun. Theor. Phys.* **67**(6), 732–742 (2017)
29. Naseri, M., Heidari, S., Gheibi, R., Gong, L.H., Raji, M.A., Sadri, A.: A novel quantum binary images thinning algorithm: a quantum version of the Hilditch's algorithm. *Optik* **131**, 678–686 (2017)
30. Heidari, S., Farzadnia, E.: A novel quantum LSB-based steganography method using the gray code for colored quantum images. *Quantum Inf. Process.* **16**(10), 242 (2017)
31. Heidari, S., Pourarian, M.R., Gheibi, R., Naseri, M., Houshmand, M.: Quantum red-green-blue image steganography. *Int. J. Quantum Infor.* **15**(05), 1750039 (2017)
32. Jiang, N., Wu, W.Y., Wang, L.: The quantum realization of Arnold and Fibonacci image scrambling. *Quantum Inf. Process.* **13**(5), 1223–1236 (2014)
33. Jiang, N., Wang, L., Wu, W.Y.: Quantum Hilbert image scrambling. *Int. J. Theor. Phys.* **53**(7), 2463–2484 (2014)
34. Jiang, N., Wang, L.: Analysis and improvement of the quantum Arnold image scrambling. *Quantum Inf. Process.* **13**(7), 1545–1551 (2014)
35. Zhou, R.G., Sun, Y.J., Fan, P.: Quantum image gray-code and bit-plane scrambling. *Quantum Inf. Process.* **14**(5), 1717–2734 (2015)
36. Anushialevi, R., et al.: Revolving of pixels and bits in pixels-plan (E) Tray encryption. *Res. J. Inf. Thechnol.* **9**, 25–31 (2017)
37. Shende, V.V., Markov, I.L.: On the CNOT-cost of Toffoli gates. arXiv preprint [arXiv:0803.2316](https://arxiv.org/abs/0803.2316) (2008)
38. Jagadeesh, P., Nagabhushan, P., Pradeep Kumar, R.: A novel image scrambling technique based on information entropy and quad tree decomposition. *Int. J. Comput. Sci. Issues* **10**(2), 1694–784 (2013)