Quantum Information Processing (2019) 18:272
https://doi.org/10.1007/s11128-019-2386-3

An encryption protocol for NEQR images based on one-particle quantum walks on a circle

Bassem Abd-El-Atty[1] · Ahmed A. Abd El-Latif[1,2] · Salvador E. Venegas-Andraca[3]

Received: 29 June 2018 / Accepted: 18 July 2019 / Published online: 26 July 2019

Abstract
Quantum walks are generalizations of random walks that have extensive applications in various fields including cryptography, quantum algorithms, and quantum networking. Discrete quantum walks can be seen as nonlinear mappings between quantum states and position probability distributions, and this mathematical property may be thought of as an imprint of chaotic behavior and consequently used to generate encryption keys. In this paper, we introduce encryption and decryption algorithms for NEQR images based on discrete quantum walks on a circle. We present full quantum circuits of proposed encryption and decryption algorithms together with digital computer simulations of most common attacks on encrypted images. Our numerical results show that our quantum image encryption and decryption scheme has high efficiency and high security with high large key space.

Keywords Discrete-time quantum walks · Quantum walks on a circle · chaotic systems · Quantum image processing · Quantum image encryption

1 Introduction

Images constitute a very popular information source for human beings. Since raw digital images can be maliciously manipulated and altered, the protection of image

✉ Salvador E. Venegas-Andraca
salvador.venegas-andraca@keble.oxon.org; svenegas@tec.mx

1 Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shebin El-Koom 32511, Egypt

2 School of Information Technology and Computer Science, Nile University, Sheikh Zayed City, Egypt

3 Tecnologico de Monterrey, Escuela de Ingenieria y Ciencias, Ave. Eugenio Garza Sada 2501, 64849 Monterrey, N.L., Mexico

data from unauthorized access has become a crucial issue studied by experts and researchers [1].

Image encryption is a widely used technique for protecting images, and it refers to transforming an image from an understandable form into an unidentifiable form [2,3].

Chaotic systems play a vital role in the development of encryption and decryption algorithms as they are used to generate encryption keys [4]. Chaotic systems can be intuitively thought as dynamical systems whose behavior, in addition to being highly sensitive to initial conditions, is so complicated that predicting it easily becomes impossible [5].

There is no unanimous formal mathematical definition of a chaotic system. However, the following definition [6,7] is generally accepted: A chaotic system must fulfill the following three properties:

1. It must be sensitive to initial conditions.
2. It must be topologically mixing.
3. It must have dense periodic orbits.

Chaotic systems can be either continuous (variables evolve continuously with time) or discrete (variables change at equally spaced time steps). In the discrete case, a more operational definition has been proposed: Some discrete-time and discrete-value chaotic systems are characterized by nonlinear maps [8].

Moreover, the mappings that describe chaotic systems are both deterministic and highly sensitive to initial conditions. As stated in [4], the deterministic nature of the equations that define chaotic systems implies both the existence and uniqueness of solutions; however and in contrast, the computability of solutions does not necessarily follow from determinism, i.e., solutions may exist and be unique, but it may be impossible to exactly calculate them using a computer. Deterministic mappings of chaotic systems with domain defined in the real number system are computationally unpredictable (this is because the trajectories of those chaotic systems are not computable), while deterministic mappings defined on finite sets are always predictable because their trajectories are eventually periodic [9].

Quantum computation can be defined as a multidisciplinary field focused on the development of computers and algorithms based on the quantum mechanical properties of nature [10]. Quantum computation is transitioning from an emerging branch of science into a mature research field, with cross-fertilizing initiatives in fields such as machine learning [11–13], military technology [14], and image processing [15,16]. Moreover, several solid results of quantum computation are increasingly attracting the attention of wider audiences of computer engineers and IT managers [17].

Quantum image processing is a subfield of quantum information focused on developing quantum algorithms and quantum protocols for capturing, manipulating, and recovering visual information [18]. This field was born with the publication of [19–22], and, in spite of being at an early development stage, it has already produced key contributions in the areas of quantum image watermarking [23–26], quantum image encryption [27–35], and quantum image steganography [36–40]. Several methods for storing and processing quantum images have been proposed, among them the Novel Enhanced Quantum Representation (NEQR) and the Flexible Representation of Quantum Images (FRQI) [15,41,42].

There is a variety of quantum/classical image encryption techniques based on chaotic systems. In [30], Gong et al. presented a quantum image encryption algorithm based on quantum controlled-NOT ($\hat{C}_{\text{not}}$) operation controlled by Chen's hyper-chaotic system. Also, Liang et al. [31] presented a quantum encryption algorithm based on generalized affine transform and quantum $\hat{C}_{\text{not}}$ operation controlled by logistic map and Tan et al. [32] presented a quantum color image encryption scheme based on the same idea of [30], while Zhou et al. [33] proposed a quantum image encryption algorithm with a four-dimensional hyper-chaotic system and iterative Arnold transforms.

Quantum walk, the quantum-mechanical counterpart of random walks, is an advanced tool for building quantum algorithms that also constitutes a universal model of quantum computation [43,44]. Traditionally, quantum walks have been employed to develop quantum algorithms focused on solving graph-related and algebraic problems (e.g., [44–48]). Additionally, discrete quantum walks have been recently thought of as a resource to create quantum image encryption algorithms, according to the following rationale: The computation of the position probability distribution of a quantum walker requires calculating probabilities out of quantum amplitudes via squaring norms of complex numbers. We may then think of a discrete quantum walk Q as a nonlinear mapping $Q : \mathcal{H} \mapsto \mathcal{P}$ where $\mathcal{H}$ is a Hilbert space in which the quantum walker lives and $\mathcal{P}$ is a set of probability distributions.

Thinking of discrete quantum walks as nonlinear mappings allows to consider them as discrete-time and discrete-value chaotic systems [8]. Further considerations to support the notion of discrete quantum walks as a type of chaotic systems are the deterministic nature of evolution via unitary operators as well as being highly sensitive to initial conditions. (The shapes of position probability distributions significantly change depending on several factors, including the initial quantum state of walkers and coins.) So, we may use discrete quantum walks as key generators for encryption algorithms [49–53].

In [51], Li et al. have proposed a quantum hash function based on controlled two-particle discrete-time quantum walks on a circle. Also, Yang et al. [49] have proposed a classical image encryption approach based on two-particle quantum walks on a circle and Yang et al. [53] designed a quantum hash function and presented its application to classical image encryption which depends on the idea of controlled quantum walks in [51].

To the best of our understanding and according to our literature review, no previous research has investigated the use of quantum walks to encrypt and decrypt images stored in quantum systems. Hereinafter, we shall use the term quantum image to refer to an image stored in a set of qubits. In the contribution presented in this manuscript, we store grayscale images in the NEQR model.

So, in this paper, we present quantum image encryption and decryption algorithms based on the chaotic behavior (in the sense of nonlinear mappings) of quantum walks on a circle with N nodes controlled by a binary message m. Our proposed quantum walk-based encryption and decryption algorithms build upon the quantum walk scheme presented in [53]. Analyses and simulation results show that our presented algorithm has high efficiency with high security.

The remainder of this paper is organized as follows: Sect. 2 is devoted to introducing some preliminary notions of quantum walks on a circle and NEQR representation

model. Section 3 presents our quantum image encryption/decryption protocol. Section 4 provides numerical analyses and digital computer simulation results of our encryption/decryption protocol on several images. Finally, concluding remarks are drawn in Sect. 5.

2 Preliminaries

2.1 Quantum walks on a circle

Quantum walks constitute a generalization of random walks in the quantum world. Originally devised as models of physical phenomena as well as advanced tools for building quantum algorithms [54–56], quantum walks have been proved to constitute a universal model of quantum computation [43,44,57]. These properties together with the appealing idea of defining the notion of algorithm based on scattering theory [58] have made quantum walks a popular field of research.

There are two kinds of quantum walks: continuous and discrete quantum walks. The former evolves via the Schrödinger equation, while the latter evolves via unitary operators [44,59–63]. In this paper, we focus on discrete-time quantum walks (QWs) as a key component of our quantum image encryption algorithm.

The basic components of a coined discrete quantum walk are a coin, a walker, evolution operators, and a set of observables. A walker is a quantum system living in a Hilbert space $\mathcal{H}_p$ with $\#(\mathcal{H}_p) = \aleph_0$ if the quantum walk runs on an unlimited line or $\#(\mathcal{H}_p) = N$ if it runs on a circle of N vertices. The coin is typically a quantum system living in a two-dimensional Hilbert space $\mathcal{H}_c$. Then, the total state of a discrete quantum walk lives in $\mathcal{H}_p \otimes \mathcal{H}_c$.

The total evolution operator $\hat{U}$ for a discrete quantum walk is given by Eq. (1):

$$\hat{U} = \hat{S}(\hat{C} \otimes \hat{I}) \tag{1}$$

where $\hat{C}$ and $\hat{S}$ are the coin operator and the shift operator, respectively.

An elementary step of a coined classical random walk consists of tossing a coin, and, depending on the outcome of the coin toss, the walker would walk one step either to the left or the right. The dynamics of a coined discrete quantum walk resemble that of a coined classical random walk.

An elementary step of a coined discrete quantum walk consists of applying, to the total quantum system (walker and coin), an evolution operator to the coin state followed by a conditional shift operator. The coin operator transforms the coin state in a superposition, and the shift operator spreads the walker state over the graph upon which the quantum walk is run (for example, over $\mathbb{Z}$ if it is an unrestricted quantum walk on a line.)

In general, an r-step discrete quantum walk can be written as

$$|\psi\rangle_{t_n} = \hat{U}^r |\psi\rangle_{t_0} \tag{2}$$

or, equivalently, as

$$|\psi\rangle_{t_r} = \sum_k [a_k|0\rangle_c + b_k|1\rangle_c]|k\rangle_p \tag{3}$$

The general matrix representation of a two-dimensional coin operator is given by Eq. (4) [44,63]:

$$C = \begin{pmatrix} \sqrt{\rho} & \sqrt{1-\rho}e^{i\omega} \\ \sqrt{1-\rho}e^{i\omega} & -\sqrt{\rho}e^{i(\omega+\phi)} \end{pmatrix} \tag{4}$$

where $0 \leq \rho \leq 1$ and the arbitrary angles $0 \leq \omega, \phi \leq \pi$. If we only focus on coin operators defined over $\mathbb{R}^2$, the general matrix representation of a two-dimensional coin operator is given by Eq. (5), where $\theta \in \mathbb{R}$ [49,50,52]:

$$C = \begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix} \tag{5}$$

The structure of the shift operator $\hat{S}$ depends on the graph the quantum walk is running on. For instance, a shift operator for a coined discrete quantum walk on an unrestricted line is given by Eq. (6)

$$\hat{S} = \sum_x (|x+1, 0\rangle\langle x, 0| + |x-1, 1\rangle\langle x, 1|) \tag{6}$$

Equation (7) is a suitable shift operator for a circle with N nodes:

$$\hat{S} = \begin{cases} |2, 0\rangle\langle 1, 0| + |N, 1\rangle\langle 1, 1| & \text{when } x = 1 \\ |1, 0\rangle\langle N, 0| + |N-1, 1\rangle\langle N, 1| & \text{when } x = N \\ |x+1, 0\rangle\langle x, 0| + |x-1, 1\rangle\langle x, 1| & \text{when } x \neq 1, N \end{cases} \tag{7}$$

The probability of finding the particle at position x after r steps can be stated as follows:

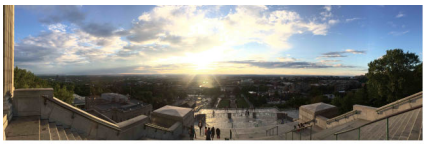
$$P(x, r) = \sum_{c\in\{0,1\}} |\langle x,c| \left(\hat{U}\right)^r |\psi\rangle_0|^2 \tag{8}$$

Since Eq. (1) defines a unitary (hence reversible) operator, the position probability distribution produced by a walker on an N-circle never reaches a uniform distribution [63].

Moreover, please note that for a circle with only N nodes, probability $P(x, r)$ is nonzero in any position if the number of steps r is greater than or equal to the number of nodes N [50,51,55].

2.2 NEQR model

Digital images are matrices of order $m \times n$ whose entries are known as pixels. Pixels can store colors, grayscale tones, or black-and-white values, as shown in Fig. 1, where we present a view of the city of Montreal in color, gray scale, and black-and-white formats.

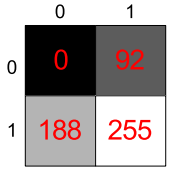


(a) Color view of the city of Montreal

(b) Grayscale view of the city of Montreal

(c) Black-and-white view of the city of Montreal

Fig. 1 Three different views (color, gray scale, and black and white) of the city of Montreal



$$|I\rangle = \frac{1}{2}(|0\rangle\otimes|00\rangle + |92\rangle\otimes|01\rangle + |188\rangle\otimes|10\rangle + |255\rangle\otimes|11\rangle)$$
$$= \frac{1}{2}(|00000000\rangle\otimes|00\rangle + |01011100\rangle\otimes|01\rangle + |10111100\rangle\otimes|10\rangle + |1111111\rangle\otimes|11\rangle)$$

Fig. 2 An example of image with size $2^1 \times 2^1$

Grayscale tones are numerically represented by integer numbers from 0 (black) to 255 (white).

Quantum images can be represented by several methods [15], among them the NEQR model [41]. The key concept behind the NEQR model is to store grayscale values in qudits. To do so, each NEQR image pixel is an element of the computational basis of $\mathcal{H}^{2^8}$, i.e., $\{|00000000\rangle, |00000001\rangle, \ldots, |11111111\rangle\}$. In order to keep track of the spatial location of each qudit, the NEQR model uses a pair of qubits $|i\rangle, |j\rangle$ as indices, where i and j are the row and column in which the accompanying qudit (grayscale quantum pixel) is located. Please note that by using the qubit indices $|i\rangle, |j\rangle$, each qudit in an NEQR image can be addressed independently of any other qudit in the quantum image. Moreover, since pixels in an NEQR image are elements of the computational basis of $\mathcal{H}^{2^8}$, then we can deterministically retrieved classical grayscale values by performing entrywise measurements using projection operators $\{|z\rangle\langle z|, z \in \{0, 1, \ldots, 255\}\}$. So, NEQR is a pertinent model for quantum grayscale image processing tasks.

Let us formally define the NEQR model. Suppose that we have an image I of size $2^n \times 2^n$. Then, the NEQR representation of I can be expressed according to Eq. (9):

$$|I\rangle = \frac{1}{2^n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1}|c_{i,j}\rangle \otimes |i\rangle|j\rangle \tag{9}$$

where $|c_{i,j}\rangle \in \{|00000000\rangle, |00000001\rangle, \ldots, |11111111\rangle\}$. Figure 2 presents an example of an NEQR image with size $2^1 \times 2^1$.

3 Proposed quantum image encryption and decryption algorithms

In this section, we present our encryption/decryption protocol for quantum images.

Please refer to Figs. 3 and 4 for a big-picture visual representation of the following steps. Moreover, Figs. 5, 6 show an example of our encryption and decryption protocol.

Encryption

Our encryption protocol consists of mixing, via $\hat{C}_{\text{not}}$ gates, two NEQR images: The first NEQR image, denoted by $|K\rangle$, contains a mask, while the second NEQR image, denoted by $|I\rangle$, stores the actual image I we want to encrypt [i.e., $|I\rangle$ is the plain image in NEQR model, Eq. (9)]. Image $|K\rangle$ is produced by the following three-step method (upper half of Fig. 3):

1. Compute a probability distribution $P = (p_1, p_2, \ldots, p_N)$ by running an r-step quantum walk on an N node circle as described in Sect. 3.1 and depicted on the left-hand side of Fig. 3. Probability distribution P can be computed via either digital computer simulations or an actual experimental implementation of a quantum walk on an N node circle:
   (a) *Digital computer simulations* In this case, an r-step quantum walk on an N node circle [Eqs. (2, 3)] is simulated on a digital computer (Sect. 3.1). Then, probability distribution P is computed by squaring the probability amplitudes of Eqs. (2, 3) following Eq. (8). This is the approach we have followed in this paper.
   (b) *Experimental realization* In this case, an r-step quantum walk on an N node circle [Eqs. (2, 3)] is experimentally run on a physical setup. After r steps, a measurement on the walker is taken. This experiment should be repeated many times in order to learn the frequency of finding the quantum walker on each node. Probability distribution P would be computed straightforwardly out of those frequencies.
2. Divide up the elements of P into n sets P_i, where each P_i is composed of n numbers (i.e., $N = n \times n$, $i \in \{1, \ldots, n\}$). Furthermore, produce a temporary order n matrix T whose columns T_i are the sets P_i. This step is encapsulated in the process 'Resize and reshape QW probability distribution to an $n \times n$ matrix' (upper side of Fig. 3).
3. We now transform matrix T into a new temporary matrix S whose entries are integer numbers between 0 and 255, i.e., we map each entry $t_{i,j} \in [0, 1]$ into an integer number $s_{i,j} \in \{0, \ldots, 255\}$, using Eq. (10)

$$s_{i,j} = \text{floor}((t_{i,j} \times 10^8) \bmod 256) \tag{10}$$

In order to compute Eq. (10), we have used the MATLAB function fix, i.e.,

$$s_{i,j} = \text{fix}((t_{i,j} \times 10^8) \bmod 256)$$

Finally, matrix S will be used as input of the NEQR protocol to produce matrix K.

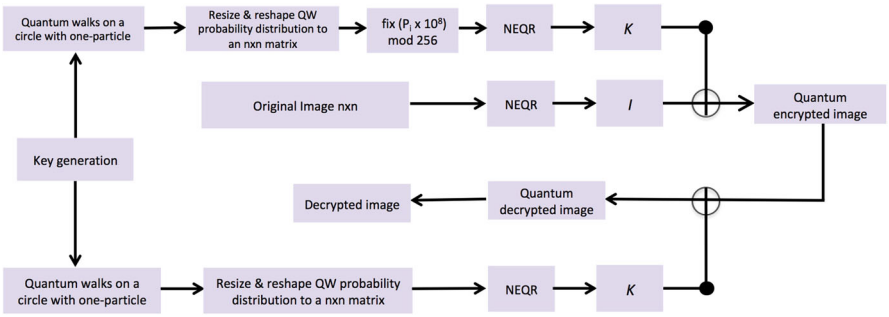


Fig. 3 The encryption and decryption processes of the proposed protocol. Here, the resize and reshape process consists of transforming probability P to an $n \times n$ (size of the original image)

The last part of our encryption protocol consists of applying $\hat{C}_{\text{not}}$ gates on grayscale qudits from image $|I\rangle$ (target qubits) using qudits from image $|K\rangle$ as control qubits. This step is explained in full detail in Sect. 3.2.

Decryption

The decryption process consists of applying a $\hat{C}_{\text{not}}$ gate on each and every qudit from the quantum encrypted image, being qudits from image $|K\rangle$ the control qubits required for these computations. Once the quantum image has been decrypted, the next and final step is to measure the NEQR quantum image in order to extract the corresponding original classical image.

In summary, the encryption and decryption processes can be described as follows:

$$\text{Quantum Encrypted Image} = C_{\text{not}}(|K\rangle, NEQR(\text{Original Image})) \tag{11}$$

$$\text{Decrypted Image} = C_{\text{not}}(|K\rangle, \text{Encrypted Image}) \tag{12}$$

where $|K\rangle$ is an NEQR image produced by transforming a probability distribution P using the NEQR model, as described above. Hereinafter, P will be known as the key of this protocol, being this name due to the fact that the role of P in our protocol resembles that of a private key in a cryptography protocol.

The key is produced by running (and measuring at a later stage) an r-step discrete quantum walk on an N node circle according to Eq. (13).

$$key = QWs(m, N, r, \alpha, \beta, \theta_1, \theta_2, \theta_3) \tag{13}$$

As described in [53], QWs are produced by running a quantum walk on a circle using three different evolution operators $\hat{U}_0, \hat{U}_1, \hat{U}_2$ presented in Eqs. (17, 18, 19). (One evolution operator is selected for each step of the quantum walk.) The key parameters are the numbers $(m, N, r, \alpha, \beta, \theta_1, \theta_2, \theta_3)$, where m is a bit string employed to select coin operators $\hat{U}_0, \hat{U}_1, \hat{U}_2$ presented in Eqs. (17, 18, 19), N is the number of nodes, r is the number of running steps of the discrete quantum walk, α and β are the amplitudes of the quantum-walk coin initial state (the walker is usually initialized as $|0\rangle$ unless required otherwise), and θ_1, θ_2, as well as θ_3 are arguments of the coin matrices presented in Eqs. (14, 15, 16). The process of generating a key by running

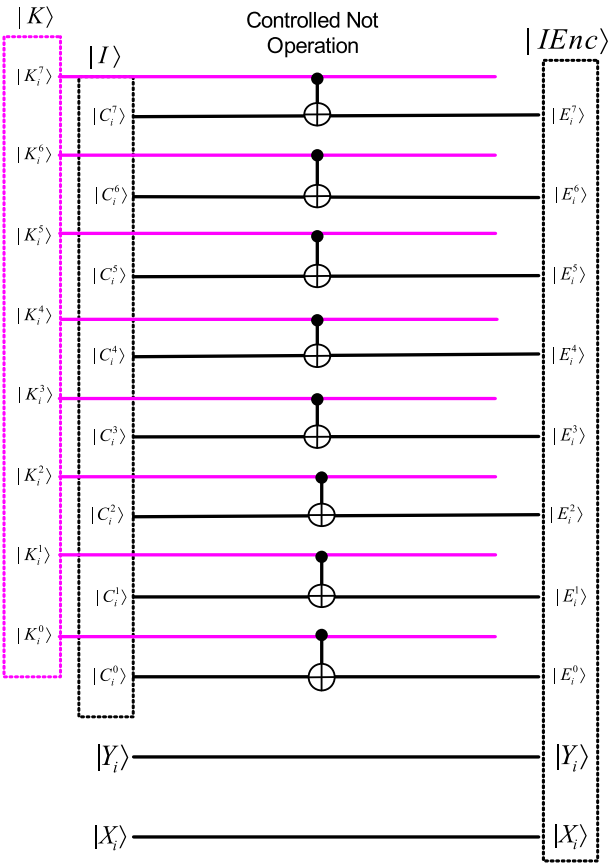


Fig. 4 Quantum circuit for the encryption algorithm

a quantum walk using key parameters (m, N, r, α, β, θ_1, θ_2, θ_3) is presented in full detail in Sect. 3.1.

Our quantum image encryption and decryption protocol is illustrated in Fig. 3, and the last circuit of the encryption protocol is presented in Fig. 4, with model example for encryption and decryption processes presented in Figs. 5 and 6, respectively.

3.1 Key generator based on quantum walks

To generate a key based on one-dimensional one-particle QWs on a circle controlled by a binary string m, we use three coins operators $\hat{C}_0$, $\hat{C}_1$, and $\hat{C}_2$ to construct evolution operators $\hat{U}_0$, $\hat{U}_1$, and $\hat{U}_2$, respectively. Matrix representations of $\hat{C}_0$, $\hat{C}_1$, $\hat{C}_2$ are presented in Eqs. (14–16).

$$C_0 = \begin{pmatrix} \cos\ \theta_1 & \sin\ \theta_1 \\ \sin\ \theta_1 & -\cos\ \theta_1 \end{pmatrix} \tag{14}$$

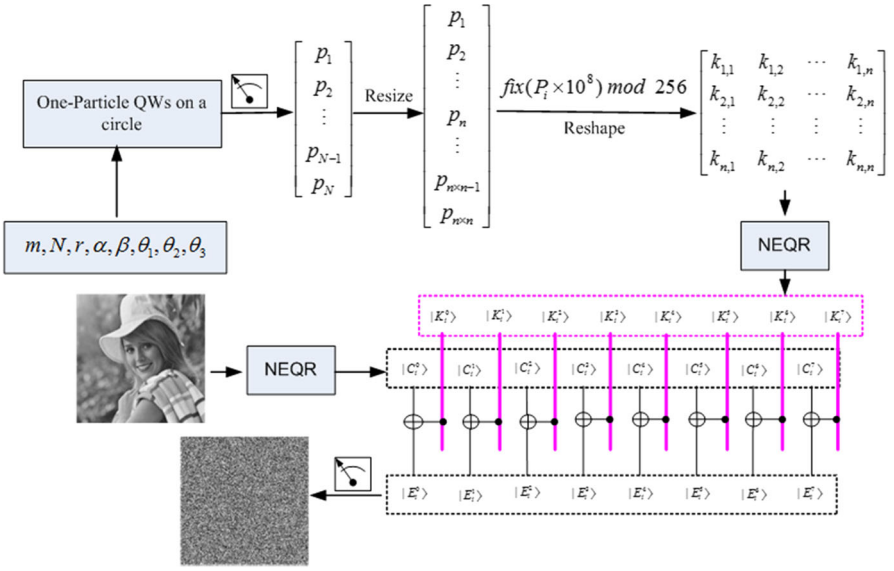


Fig. 5 A model example of the encryption algorithm. By running quantum walks on a circle with key parameters (m, N, r, α, β, θ_1, θ_2, θ_3), produce a probability vector P, then resizing it to one column with length $n \times n$ (size of the original image), thereafter reshape it to the size of the original image, then used as quantum controlled-NOT image $|K\rangle$ after converted to integer values. The plain image (Elaine, in this example) transformed to the quantum state $|I\rangle$ and encrypted by applying the C_{not} gate controlled by $|K\rangle$ (control qubit)

$$C_1 = \begin{pmatrix} \cos\ \theta_2 & \sin\ \theta_2 \\ \sin\ \theta_2 & -\cos\ \theta_2 \end{pmatrix} \tag{15}$$

$$C_2 = \begin{pmatrix} \cos\ \theta_3 & \sin\ \theta_3 \\ \sin\ \theta_3 & -\cos\ \theta_3 \end{pmatrix} \tag{16}$$

where $\theta_1, \theta_2, \theta_3 \in \{0, 2\pi\}$.

Moreover, the quantum walk evolution operators $\hat{U}_0, \hat{U}_1, \hat{U}_2$ are presented in Eqs. (17, 18, 19).

$$\hat{U}_0 = \hat{S}(\hat{I} \otimes \hat{C}_0) \tag{17}$$

$$\hat{U}_1 = \hat{S}(\hat{I} \otimes \hat{C}_1) \tag{18}$$

$$\hat{U}_2 = \hat{S}(\hat{I} \otimes \hat{C}_2) \tag{19}$$

According to the quantum walk-based key generation protocol presented in [53], quantum walk evolution operators $\hat{U}_0, \hat{U}_1, \hat{U}_2$ are selected in agreement with the following procedure: let m and r (two of the key parameters presented in Eq. (13)) be a bit string (i.e., a concatenation of 0s and 1s) and the total number of steps of the quantum walk, respectively. Then, the evolution operator $\hat{U}_i^j$ of each step jth ($j \in \{1, \ldots r\}$) of the quantum walk will be selected according to the rule presented in Eq. (20):

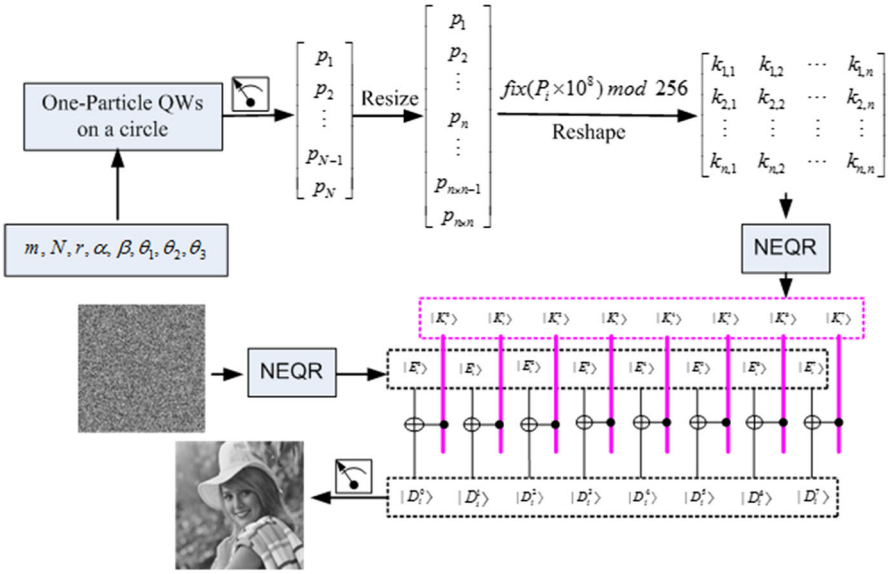


Fig. 6 A model example of the decryption algorithm. By running quantum walks on a circle with key parameters $(m, N, r, \alpha, \beta, \theta_1, \theta_2, \theta_3)$, produce a probability vector P, then resizing it to one column with length $n \times n$ (size of the encrypted image), thereafter reshape it to the size of the encrypted image, then used as quantum controlled-NOT image $|K\rangle$ after converted to integer values. The encrypted image transformed to the quantum state $|IEnc\rangle$ and decrypted by applying the $\hat{C}_{\text{not}}$ gate controlled by $|K\rangle$

$$\hat{U}_i^j = \begin{cases} \hat{U}_0, & \text{if } m_j = 0 \text{ (i.e., the } j\text{th bit of } m \text{ is equal to 0)} \\ \hat{U}_1, & \text{if } m_j = 1 \text{ (i.e., the } j\text{th bit of } m \text{ is equal to 1)} \\ \hat{U}_2, & \text{if the QW step number } j \text{ is greater than the length of } m. \end{cases} \tag{20}$$

For example, let $m = 01101$ and $r = 7$. Then, reading m from left to right (i.e., $m_0 = 0, m_1 = 1, m_3 = 1, m_4 = 0, m_5 = 1$), we shall run a seven-step quantum walk using the evolution operators presented in Eq. (21):

$$|\psi\rangle_7 = \hat{U}_{2(r>\#(m))}\hat{U}_{2(r>\#(m))}\hat{U}_{1(m_5=1)}\hat{U}_{0(m_4=0)}\hat{U}_{1(m_3=1)}\hat{U}_{1(m_2=1)}\hat{U}_{0(m_1=0)}|\psi\rangle_0 \tag{21}$$

So, generating a key in our protocol consists of selecting key parameters $(m, N, r, \alpha, \beta, \theta_1, \theta_2, \theta_3)$ for running a one-particle QWs on a circle with N nodes in order to produce a probability distribution P with size N (i.e., the key):

$$P = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{N-1} \\ p_N \end{bmatrix}$$

As stated at the beginning of this section, P can be produced either by computation or experimental realization.

As previously stated in this paper, m is a bitstring employed to select coin operators $\hat{U}_0, \hat{U}_1, \hat{U}_2$ presented in Eqs. (17, 18, 19), N is the number of nodes, r is the number of running steps of the discrete quantum walk, α and β are the amplitudes of the quantum-walk coin initial state (hence, the initial state of the coin is $|\psi\rangle_0 = \alpha|0\rangle + \beta|1\rangle$. The initial state of the quantum walker is $|0\rangle$ unless otherwise stated), and θ_1, θ_2, and θ_3 are arguments of the coin matrices presented in Eqs. (14, 15, 16).

3.2 Quantum image encryption procedure

Our proposal for image encryption consists of the following steps.

1. Produce NEQR images $|I\rangle$ $|K\rangle$ (Eqs. (22, 23)) using the NEQR model for quantum image representation (Sect. 2.2).

$$|K\rangle = \frac{1}{2^n} \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} |k_{i,j}^7 k_{i,j}^6 \dots k_{i,j}^1 k_{i,j}^0\rangle \otimes |i\rangle|j\rangle \tag{22}$$

$$|I\rangle = \frac{1}{2^n} \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} |c_{i,j}^7 c_{i,j}^6 \dots c_{i,j}^1 c_{i,j}^0\rangle \otimes |i\rangle|j\rangle \tag{23}$$

   where $k_{i,j}^g, c_{i,j}^g \in \{0, 1\}$ with $g \in \{0, \dots 7\}$.
2. The encrypted quantum image is produced by applying entrywise $\hat{C}_{\text{not}}$ gates to $|I\rangle$, the NEQR representation of the original classical image, using entries of $|K\rangle$ as control qubits. Let us denote the quantum encrypted image by $|X\rangle$, then each entry $|x_{i,j}^7 x_{i,j}^6 \dots x_{i,j}^0\rangle \otimes |i\rangle|j\rangle$ is computed according to Eq. (24)

$$\begin{aligned} |x_{i,j}^7 x_{i,j}^6 \dots x_{i,j}^0\rangle \otimes |i\rangle|j\rangle &= \hat{C}_{\text{not}}^{\otimes 8} |k_{i,j}^7 k_{i,j}^6 \dots k_{i,j}^0\rangle |c_{i,j}^7 c_{i,j}^6 \dots c_{i,j}^0\rangle \otimes |i\rangle|j\rangle \\ &= |C_{\text{not}}(k_{i,j}^7, c_{i,j}^7) C_{\text{not}}(k_{i,j}^6, c_{i,j}^6) \dots C_{\text{not}}(k_{i,j}^0, c_{i,j}^0)\rangle \otimes |i\rangle|j\rangle \end{aligned} \tag{24}$$

   for each pair of indices $|i\rangle, |j\rangle$. So,

$$|X\rangle = \frac{1}{2^n} \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} |x_{i,j}^7 x_{i,j}^6 \dots x_{i,j}^0\rangle \otimes |i\rangle|j\rangle. \tag{25}$$

3.3 Quantum image decryption procedure

The decryption procedures of the proposed algorithm consist of the following steps.

1. Take the encrypted image

$$|X\rangle = \frac{1}{2^n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1}|x_{i,j}^7 x_{i,j}^6 \ldots x_{i,j}^0\rangle \otimes |i\rangle|j\rangle$$

and the control image

$$|K\rangle = \frac{1}{2^n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1}|k_{i,j}^7 k_{i,j}^6 \ldots k_{i,j}^1 k_{i,j}^0\rangle \otimes |i\rangle|j\rangle$$

2. The decrypted quantum image $|I\rangle$ is retrieved by applying entrywise $\hat{C}_{\text{not}}$ gates to $|X\rangle$, the encrypted NEQR image using entries of $|K\rangle$ as control qubits. Entries $|c_{i,j}^7 c_{i,j}^6 \ldots c_{i,j}^0\rangle \otimes |i\rangle|j\rangle$ are computed according to Eq. (26)

$$\begin{aligned}|c_{i,j}^7 c_{i,j}^6 \ldots c_{i,j}^0\rangle \otimes |i\rangle|j\rangle &= \hat{C}_{\text{not}}^{\otimes 8}|k_{i,j}^7 k_{i,j}^6 \ldots k_{i,j}^0\rangle|x_{i,j}^7 x_{i,j}^6 \ldots x_{i,j}^0\rangle \otimes |i\rangle|j\rangle \\ &= |C_{\text{not}}(k_{i,j}^7, x_{i,j}^7)C_{\text{not}}(k_{i,j}^6, x_{i,j}^6)\ldots C_{\text{not}}(k_{i,j}^0, x_{i,j}^0)\rangle \otimes |i\rangle|j\rangle \end{aligned} \quad (26)$$

4 Numerical simulations based on classical computers

To simulate the proposed quantum encryption algorithm with quantum walks on a circle, a laptop with Intel $Core^{TM}$ i5 CPU 2.50 GHz and 6 GB RAM equipped with MATLAB software R2017a is used to perform unitary transformations on quantum walks and quantum operations on quantum images. Lena, Elaine, baboon, cameraman, and boats are five images which are used as test images with size (256×256) (Fig. 7). The key parameters used to run the quantum walks on a circle to generate the key sequence are (m=[0110 1000 1000 1010 1111 1000 1111 1000], $N = 257$, $r = 770$, $\alpha = 1$, $\beta = 0$, $\theta_1 = \pi/3$, $\theta_2 = \pi/4$ and $\theta_3 = \pi/6$).

4.1 Statistical and differential analysis

The correlations between original and encrypted images can be determined by statistical analysis. In this manner, the encrypted image must be totally different from the original one. To ensure the efficiency of the proposed quantum image encryption algorithm against statistical attacks, the following statistical and differential analyses have been performed on the proposed scheme to figure out if the encrypted image releases any data about the original one or not: correlation of adjacent pixels, number of pixel change rate, histogram analysis, Shannon entropy, and spectrum analysis.

4.1.1 Correlation of adjacent pixels

The correlation coefficient of adjacent pixels C_{xy} is used to measure the relationship between the plain image and its cipher image. The correlation coefficients C_{xy}

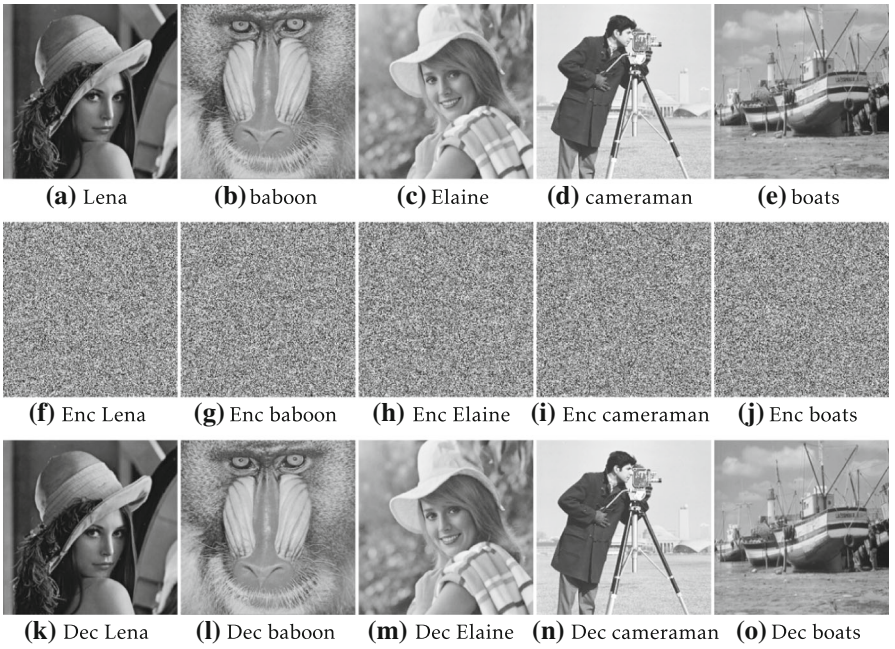


Fig. 7 The test results of the proposed quantum image encryption algorithm. The first row is the original images of Lena, baboon, Elaine, cameraman, and boats. The second row is the encrypted images of the first row. The third row is the decrypted images of the second row utilizing the key parameters used in the encryption process

of normal images are close to 1 in each direction (which implies that neighboring pixels exhibit high correlation), while in an encrypted image with a good encryption approach should close to 0 (no relation between neighboring pixels). To calculate the correlations coefficients in each direction in the encrypted and original images, we selected randomly 10,000 pairs of neighboring pixels in each direction.

$$C_{xy} = \frac{\sum_{i=1}^{M} (x_i - \bar{x})\,(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{M} (x_i - \bar{x})^2 \sum_{i=1}^{M} (y_i - \bar{y})^2}} \tag{27}$$

where x_i, y_i refer to the values of adjacent pixels and M refers to the total number of adjacent pixel pairs in each direction.

To test our protocol, we have computed the correlation coefficient of adjacent grayscale values from plain and encrypted images. Results are shown in Table 1, where we state the results of C_{xy} for two pairs of the encrypted image and their corresponding original one, and Fig. 8 shows the correlation distribution of two neighboring pixels in each direction for Lena image. Since C_{xy} values of the encrypted images are close

Table 1 Coefficients of correlations for adjacent pixels

Image	Direction		
	Horizontal	Vertical	Diagonal
Lena	0.9767	0.9650	0.9460
Enc (Lena)	− 0.0003	− 0.0013	− 0.0066
Elaine	0.9759	0.9722	0.9527
Enc (elaine)	− 0.0032	− 0.0005	− 0.0063
baboon	0.8352	0.8767	0.7963
Enc (baboon)	− 0.0078	− 0.0073	0.0073
cameraman	0.9555	0.9241	0.8990
Enc (cameraman)	− 0.0126	− 0.0002	− 0.0047
boats	0.9444	0.9294	0.8830
Enc (boats)	− 0.0096	− 0.0011	− 0.0031

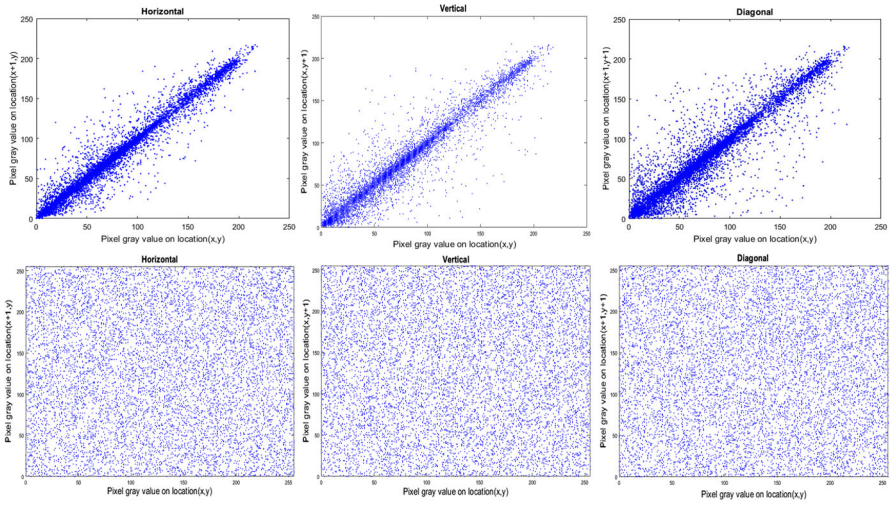


Fig. 8 Correlation distribution of two adjacent horizontal, vertical, and diagonal pixels for Lena image. The first row and the second row for the original and encrypted image, respectively

to 0, no useful information can be obtained about the original image by analysis on the correlations of neighborhood pixels.

4.1.2 Number of pixel change rate

To measure the effect of changing pixel values in the original image on its corresponding encrypted image, two tools are used: The first tool is the number of pixels change rate (NPCR), and the other is the unified average changing intensity (UACI). The NPCR and UACI can be defined as follows.

$$\mathrm{NPCR} = \frac{\sum_{i,j} D(i,j)}{M} \times 100\%, \quad D(i,j) = \begin{cases} 0 & \text{if } X(i,j) = Y(i,j) \\ 1 & \text{if } X(i,j) \neq Y(i,j) \end{cases} \tag{28}$$

Table 2 UACI and NPCR test results

Image	NPCR%	UACI%
Lena	99.58	34.08
Baboon	99.58	27.31
Elaine	99.58	28.46
Cameraman	99.58	34.76
Boats	99.58	28.31

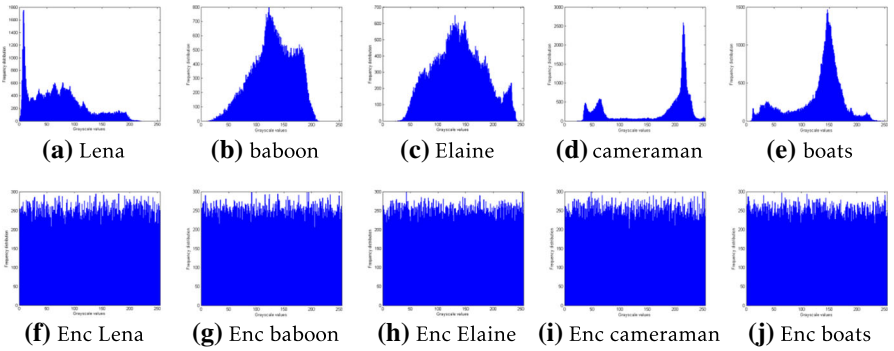


Fig. 9 Histograms of original and encrypted images. The first row is the histogram of original images for Lena, baboon, Elaine, cameraman, and boats. The second row is the histogram of the encrypted images

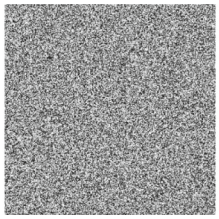
$$\mathrm{UACI} = \frac{1}{M}\left(\sum_{i,j}\frac{|X(i,j) - Y(i,j)|}{2^N - 1}\right) \times 100\% \tag{29}$$

where N refers to the number of bits used to represent the pixel values and M refers to the total number of pixels used in the image. The NPCR and UACI values are presented in Table 2, and the NPCR values for all tested images are 99.58%, so that the proposed quantum algorithm is very sensitive to small pixel changes in the original image.

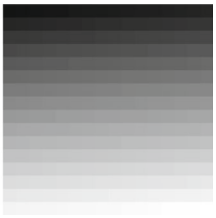
4.1.3 Histogram analysis

The histogram of an image is defined as a histogram whose bins are pixel values. Hence, in the case of grayscale images, histograms reflect the distribution of tonal values from 0 to 255. Histogram analysis is a vital tool to evaluate the performance of an encryption algorithm as it reflects the frequency distribution of pixel values in an image. Any efficient encryption algorithm should ensure the uniform histograms of different encrypted images to resistance against statistical attacks. Figure 9 gives the histograms of several original images which are different from each other, while the histograms of their corresponding encrypted images are uniform with each other. Hence, our proposed quantum image encryption protocol could resist histogram analysis attacks.

(a) Cameraman



(b) Uniform Random Noise



(c) Grayscale Stripes

Fig. 10 In **a**, light grayscale pixels are more abundant than dark grayscale pixels. This property, together with the spatial distribution of pixels across this image, allows us to identify its components. Pixels from **b** have been produced using a uniform random noise source. Pixels from **c** have the same distribution of gray levels, but their spatial correlation is different from that of (**b**). **b**, **c** were taken from http://www.johnloomis.org/ece563/notes/basics/entropy/entropy.html

4.1.4 Shannon entropy analysis

In information theory, entropy is the average amount of information obtained by observing a source output [64] and it is defined in Eq. (30)

$$E(X) = -\sum_{j=1}^{n} p(a_j)\log(p(a_j)) \tag{30}$$

where $\{a_1, \ldots, a_n\}$ is the set of symbols produced by the source (i.e., the source output) and $X = \{p(a_1), \ldots, p(a_n)\}$ is a probability distribution of symbols $\{a_1, \ldots, a_n\}$. As the magnitude of $E(X)$ increases, more uncertainty is associated with the source. If source symbols $\{a_1, \ldots, a_n\}$ are equally probable, $E(X)$ is maximized.

The notion of entropy is adopted in image cryptoanalysis by associating it with the randomness of pixel value distribution in an image. Here, we refer only to the frequency of finding pixel values in an image, regardless of the *spatial* distribution of pixels across an image.

The intuition of entropy in image cryptoanalysis is as follows: The distribution of pixel values in an image is key in order to produce visual representations of our world that allow humans to easily extract useful information out of that image. Now, if the probability of finding a pixel value is the same for all possible pixel values, i.e., if pixel values are uniformly distributed, we are basically in front of an image which is either random or with very simple patterns that are unlikely to provide useful information to humans.

For instance, let us analyze the images presented in Fig. 10. In image (a), the relative abundance of light grayscale pixels with respect to dark grayscale pixels, together with the spatial distribution of both pixel sets, allows us to distinguish the components of a man, a camera, and the elements in the background. As for images (b) and (c), pixels of both images have the same grayscale frequency but different spatial distribution. In particular, image b) has been produced using a random noise source. An encryption protocol that transforms an arbitrary image into an image similar to Fig. 10b would

Table 3 Information entropy of original and encrypted images

Image	Original image	Encrypted image
Lena	7.3275	7.9967
Baboon	7.2269	7.9973
Elaine	7.4878	7.9976
Cameraman	6.9046	7.9967
Boats	7.1583	7.9973

be highly regarded because it would leave no visible trace of the information content of the original image.

Information entropy, presented in Eq. (31), is a statistical measure of the pixel values distribution in an image.

$$E(X) = -\sum_{i=1}^{2^L-1} p(x_i) \log_2 (p(x_i)) \tag{31}$$

where x_i are pixel values and $p(x_i)$ is the probability distribution that is associated with the frequency of each pixel value in the image. In our case, x_i are grayscale values and $p(x_i)$ is the probability distribution that results from the relative frequency of each grayscale pixel value.

The highest information entropy value for grayscale images is 8, which corresponds to having the same frequency (i.e., the same probability of occurrence) for each grayscale value. Since there are $256 = 2^8$ different grayscale values x_i and $p(x_i) = \frac{1}{2^8}$ corresponds to maximizing Eq. (31), then

$$E(X) = -\sum_{i=1}^{2^8} p(x_i) \log_2(p(x_i)) = -\sum_{i=1}^{2^8} \frac{1}{2^8} \log_2 \left(\frac{1}{2^8}\right) = -\sum_{i=1}^{2^8} \frac{-8}{2^8} = 8$$

Table 3 presents the values of information entropy of five images (Lena, baboon, Elaine, cameraman, and boats) for both original and encrypted versions. The information entropy values reported in Table 3 for encrypted images show that our protocol has a good performance as information entropy values in all five cases are greater than 7.99, i.e., very close to 8.

4.1.5 Spectrum analysis

Spectrum analysis is another important tool to evaluate encrypted images, and it consists of studying the properties of images via Fourier transform. Let $f(x, y)$ be an image, then the discrete Fourier transform $F(u, v)$ of image $f(x, y)$ is given by

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi i (\frac{ux}{N} + \frac{vy}{N})}$$

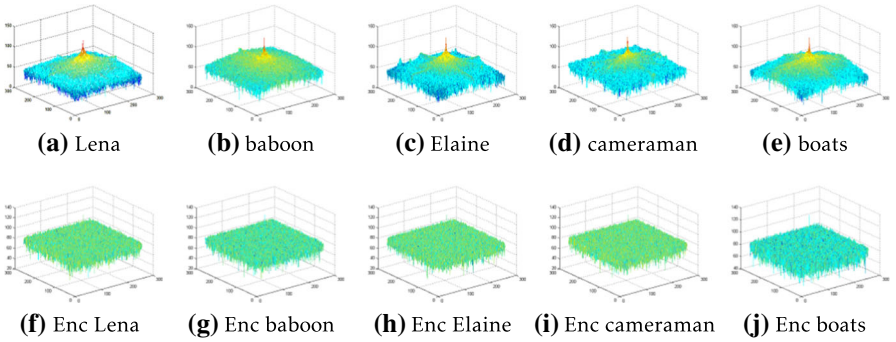


Fig. 11 Spectrum analysis of original and encrypted images. The first row is the spectrum analysis of original images for Lena, baboon, Elaine, cameraman, and boats. The second row is the spectrum analysis of the encrypted images

Table 4 The mean standard deviation values of original and encrypted images

Image	Original image	Encrypted image
Lena	41.0486	73.8650
Baboon	35.3705	73.8205
Elaine	42.1200	73.9132
Cameraman	50.5653	73.7061
Boats	42.4630	73.9906

The amplitude spectrum of $F(u, v) = F_{\text{real}}(u, v) + iF_{\text{im}}(u, v)$ is given by

$$||F(u, v)|| = \sqrt{(F_{\text{real}}(u, v))^2 + (F_{\text{im}}(u, v))^2} \tag{32}$$

Spectrum analysis is used to evaluate the robustness of encryption algorithms against statistical attacks. Any efficient encryption algorithm should have the spectrum amplitude of encrypted images nearly uniform, and their corresponding original images are different. Figure 11 shows the spectrum analysis for the encrypted images which are nearly uniform and different from their original ones. To ensure the uniform distributions for the encrypted images, Table 4 states the values of mean standard deviation which is close to 73.8 for all encrypted images. The results prove the good distribution of pixels in encrypted image. Thus, the encryption algorithm is reasonably secure against spectrum analysis attack.

4.2 Key space analysis

A requisite for a robust image encryption algorithm is to have access to a (very) large number of keys, i.e., a large key space. This is because having access to a modest number of keys would eventually force us to recycle keys and that is a typical weakness to be used by hackers (in brute-force attacks, for instance) in order to impair cryptography protocols and information technology systems in general.

In our protocol, keys are produced by running quantum walks with key parameters $(N, r, \alpha, \beta, \theta_1, \theta_2, \theta_3)$ according to the procedure presented in Subsec. (3.1) as well as in the beginning of Sec. (3). Let us suppose that we run r-step quantum walks, where $r \in \{r_1, r_2, \ldots, r_n\}$ and r_1 is a big enough number for key production (more about it in the following lines). Then, by careful choices of parameters $\alpha, \beta, \theta_1, \theta_2, \theta_3$, we could produce 2^{r_i} quantum walks for each $\{r_1, r_2, \ldots, r_n\}$.

Bitcoin private keys are 256 bit long [65]; hence, the greatest decimal value that corresponds to a Bitcoin private key is $2^{256} - 1$, roughly 1.157×10^{77}. So, letting $r_1 = 256$ would be a reasonable choice for current standards.

Let us now calculate the cardinality of a key space with keys produced by r-step quantum walks, where $r \in \{r_1, r_2, \ldots, r_n\}$.

Since

$$\sum_{i=r_1}^{r_n} 2^i = \sum_{i=1}^{r_n} 2^i - \sum_{i=1}^{r_1-1} 2^i \quad \text{and} \quad \sum_{i=0}^{n-1} 2^i = 2^n - 1$$

then

$$\sum_{i=1}^{r_1-1} 2^i = 2^{r_1} - 2 \tag{33}$$

$$\sum_{i=1}^{r_n} 2^i = 2^{r_n+1} - 2 \tag{34}$$

Then, combining Eqs. (33) and (34), we find that

$$\sum_{i=r_1}^{r_n} 2^i = 2^{r_n+1} - 2^{r_1} \tag{35}$$

For instance, if $r_1 = 300$ and $r_n = 700$, then the cardinality of the key space would be

$$\text{Cardinality of the key space } = 2^{701} - 2^{500} = 2^{500}(2^{201} - 1) = O(2^{701}) \tag{36}$$

which is large enough for a solid cryptography protocol.

Table 5 presents a comparison of key spaces for the proposed quantum image encryption algorithm and other classical/quantum image encryption algorithms [30–33,49,53]. Table 5 shows that the key space of our proposed encryption protocol performs well with respect to other encryption algorithms.

4.3 Key sensitivity analysis

One of the vital and essential tools for any secure quantum image encryption algorithms is key sensitivity, which is known as the sensitivity of the secret key to encrypt and

Table 5 Key space of our algorithm and other related algorithms

Algorithm	Key generator	Key space
Our algorithm	1D one-particle quantum walks on a circle to encrypt quantum image with key parameters $(m, N, r, \alpha, \beta, \theta_1, \theta_2, \theta_3)$.	The cardinality of our key space is given by $2^{r_n+1} - 2^{r_1} = O(2^{r_n})$ for large enough r_n, r_1. For instance, $r_n = 700$ and $r_n = 500$ make the cardinality of our key space equal to $O(2^{701})$
Gong et al. [30]	Chen's hyper-chaotic system to encrypt quantum image	The main key parameters (x_0, y_0, z_0, w_0) are used to run Chen's hyper-chaotic system. Then, transform each generated sequence into integer sequences $x_i^* = \lvert fix((x_i - fix(x_i)) \times 10^{14}) \rvert mod \ \ 256$. Each of key parameters has key space 10^{14}. So the key space of whole system is 10^{56} and depends on the power of 10^{14}
Tan et al. [32]	Chen's hyper-chaotic system to encrypt quantum color image	The main key parameters (x_0, y_0, z_0, w_0) are used to run Chen's hyper-chaotic system. Then, transform each generated sequence into integer sequences $x_i^* = \lvert fix((x_i - fix(x_i)) \times 10^{15}) \rvert mod \ \ 256$. Each of key parameters has key space 10^{15}. So, the key space of whole system is 10^{60} and depends on the power of 10^{15}
Liang et al. [31]	Logistic map to encrypt quantum image	Key parameters (L_0, δ) are used to run logistic map. Then, transform the generated sequence into integer sequences $L_i^* = \lvert floor(L_i \times 2 \wedge 8) \rvert mod \ \ 256$. The key parameters L_0 have key space 2^8. In addition to the decimal points of δ
Zhou et al. [33]	4D hyper-chaotic system to encrypt quantum image	The main key parameters (x_0, y_0, z_0, w_0) are used to run 4D hyper-chaotic system. The key space of whole system is 10^{58}
Yang et al. [49]	1-D two-particle quantum walks on a circle to encrypt classical image	The key parameters $(N, r, \alpha_1, \beta_1, \alpha_2, \beta_1, \theta)$ are used to run QWs. The key space for key parameters is 2^{325}
Yang et al. [53]	1-D two-particle quantum walks on a circle to encrypt classical image	The key parameters $(m, N, \alpha_1, \beta_1, \alpha_2, \beta_1)$ are used to run QWs. The key space for key parameters and initial states is 2^{325}

decrypt effects. For a good secure encryption algorithm, any tiny changes in key parameters lead to huge changes in the cipher image. Let plain image P encrypted twice into cipher images C_1 and C_2 by two key parameters K_1 and K_2 with tiny changes, respectively. Therefore, the key sensitivity for encryption process can be expressed as follows:

$$KS_E = \frac{\text{diff}\,(C_1, C_2)}{Number\ of\ \ pixels} \times 100\% \tag{37}$$

where diff (C_1, C_2) refers to the total number of different pixels between two images C_1 and C_2. Figure 12 shows the key encryption sensitivity for 120 images with size

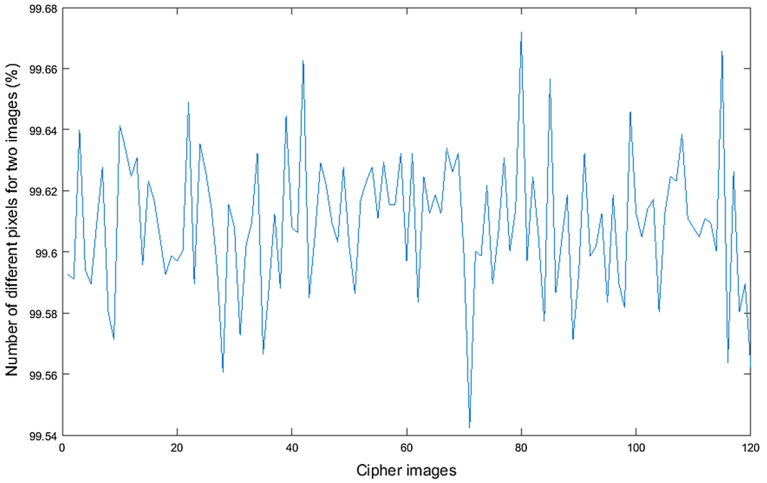


Fig. 12 Key encryption sensitivity for 120 plain images. Every plain image encrypted twice into cipher images C_1 and C_2 by two key parameters K_1 and K_2 with tiny changes, respectively. Then, count the number of different pixels between C_1 and C_2

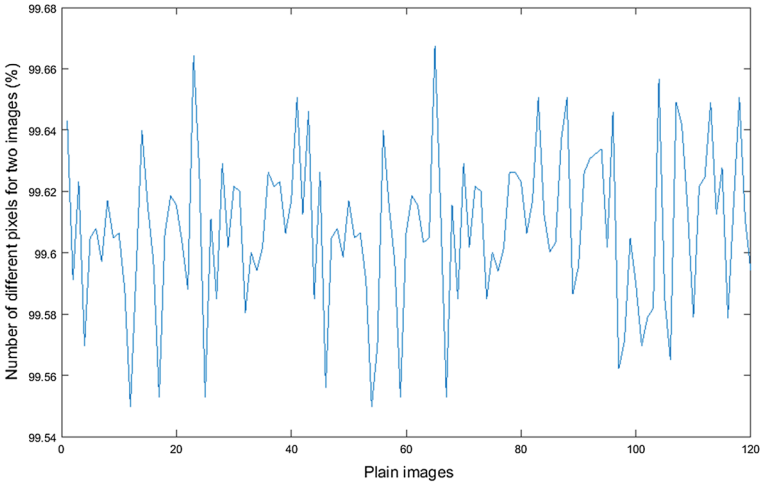
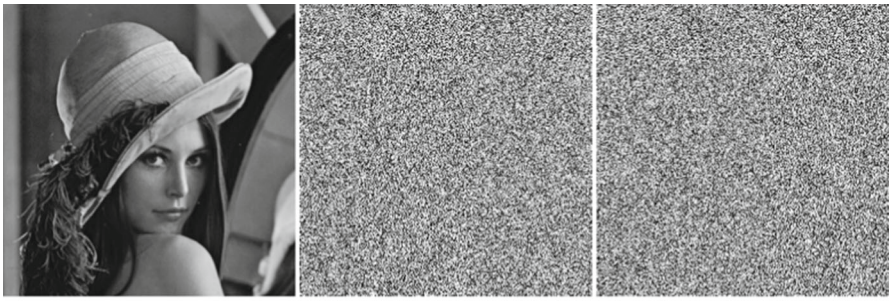


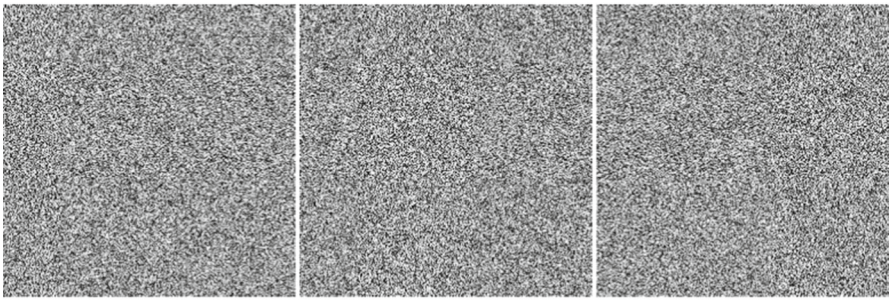
Fig. 13 Key decryption sensitivity for 120 cipher images. Every cipher image decrypted twice into plain images P_1 and P_2 by two key parameters K_1 and K_2 with tiny changes, respectively. Then, count the number of different pixels between P_1 and P_2

256×256, and the average for key sensitivity is 99.6090%. On the other hand, let cipher image C decrypted twice into plain images P_1 and P_2 by two key parameters K_1 and K_2 with tiny changes, respectively. Therefore, the key sensitivity for decryption process can be computed as follows:

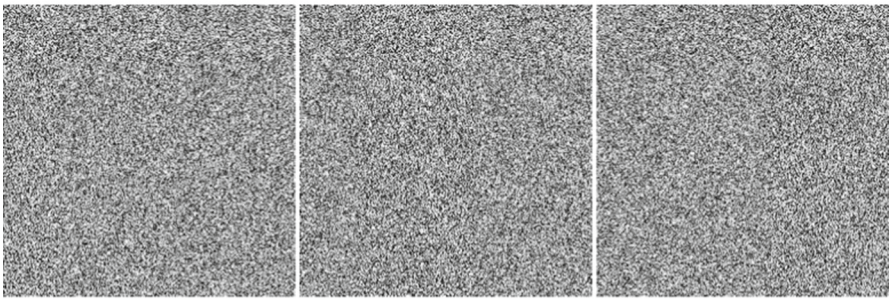
$$KS_D = \frac{\text{diff}\,(P_1, P_2)}{Number\ of\ \ pixels} \times 100\% \tag{38}$$



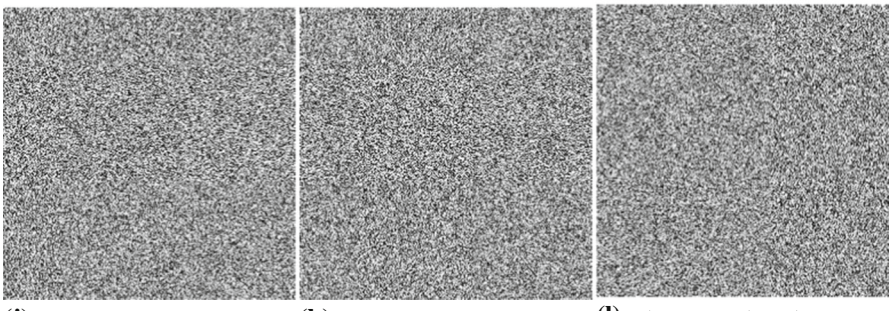
(a) The correct keys **(b)** The correct keys but r=769 **(c)** The correct keys but N=259



(d) The correct keys but $\theta_1 = 65°$ **(e)** The correct keys but $\theta_2 = 40°$ **(f)** The correct keys but $\theta_3 = 35°$



(g) The correct keys but change the first bit only of the binary string from 0 to 1 **(h)** The correct keys but change the second bit only of the binary string from 1 to 0 **(i)** The correct keys but add the bit 1 at the end of the binary string



(j) The correct keys but remove the last bit of the binary string **(k)** The correct keys but $\alpha = 0, \beta = 1$ **(l)** The correct keys but $\alpha = 1/\sqrt{2}, \beta = 1/\sqrt{2}$

Fig. 14 Decrypted image Lena with several keys

Figure 13 shows the key decryption sensitivity for 120 images with size 256×256, and the average is 99.6077%. Figures 12 and 13 show that our algorithm has high key sensitivity. For more illustration to evaluate the key sensitivity of the proposed algorithm, more tests were carried out to decrypt the cipher image Lena with several keys as shown in Fig. 14.

5 Concluding remarks

This paper has introduced a new quantum NEQR image encryption and decryption protocol based on controlled one-dimensional one-particle quantum walks on a circle, being the latter a source of large encryption and decryption keys. In addition to full encryption and decryption quantum circuits, we have presented simulations and numerical analyses that demonstrate that our proposed quantum encryption algorithm is secure against most known attack techniques.

Acknowledgements All authors would like to thank Professor Dan Li at Nanjing University of Aeronautics and Astronautics and Professor Mohamed Amin at Menoufia University for their insightful comments and valuable feedback. SEVA gratefully acknowledges the unconditional support of his family as well as the financial support of Tecnologico de Monterrey, Escuela de Ingenieria y Ciencias and CONACyT (SNI number 41594 as well as Fronteras de la Ciencia project number 1007).

References

1. Patel, K.D., Belani, S.: Image encryption using different techniques: a review. Int. J. Emerg. Technol. Adv. Eng. **1**(1), 30 (2011)
2. El-Latif, A.A.A., Li, L., Zhang, T., Wang, N., Song, X., Niu, X.: Digital image encryption scheme based on multiple chaotic systems. Sens. Imaging Int. J. **13**(2), 67 (2012)
3. El-Latif, A.A.A., Li, L., Niu, X.: A new image encryption scheme based on cyclic elliptic curve and chaotic system. Multimed. Tools Appl. **70**(3), 1559 (2014)
4. Kocarev, L., Shiguo, L. (eds.): Chaos-Based Cryptography. Theory, Algorithms and Applications. Studies on Computational Intelligence. Springer, New York (2011)
5. Sobottka, M., de Oliveira, L.: Periodicity and predictability in chaotic system. Am. Math. Mon. **113**(5), 415 (2006)
6. Devaney, R.L.: An Introduction to Chaotic Dynamical Systems, 2nd edn. Addison-Wesley, Boston (1990)
7. Banks, J., Dragan, V., Jones, A.: Chaos: A Mathematical Introduction. Cambridge University Press, Cambridge (2003)
8. Alvarez, G., Li, S.: Some basic cryptographic requirements for chaos-based cryptosystems. Int. J. Bifurc. Chaos **16**(8), 2129 (2006)
9. Kocarev, L.: Chaos-based cryptography: a brief overview. IEEE Circuits Syst. Mag. **1**(3), 6 (2001)
10. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University, Cambridge (2000)
11. Wittek, P.: Quantum Machine Learning. Academic Press, Cambridge (2014)
12. Schulda, M., Sinayskiy, I., Petruccione, F.: An introduction to quantum machine learning. Contemp. Phys. **56**(2), 172 (2015)
13. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature **549**(7671), 195–202 (2017)
14. Lanzagorta, M.: Quantum Radar. Morgan and Claypool, San Rafael (2011)
15. Yan, F., Iliyasu, A.M., Venegas-Andraca, S.E.: A survey of quantum image representations. Quantum Inf. Process. **15**, 1 (2016)

16. Abura'ed, N., Khan, F., Bhaskar, H.: Advances in the quantum theoretical approach to image processing applications. ACM Comput. Surv. **49**(4), 1–49 (2017)
17. Abd-El-Atty, B., Venegas-Andraca, S.E., El-Latif, A.A.A.: Quantum information protocols for cryptography. In: Hassanien, A.E., Elhoseny, M., Kacprzyk, J. (eds.) Quantum Computing: An Environment for Intelligent Large Scale Real Application, pp. 3–23. Springer, Cham (2018)
18. Yan, F., Chen, K., Venegas-Andraca, S., Zhao, J.: Quantum image rotation by an arbitrary angle. Quantum Inf. Process. **16**, 282 (2017)
19. Vlasov, A.: Quantum Computations and Image Recognition. arXiv:quant-ph/9703010 (1997)
20. Beach, G., Lomont, C., Cohen, C.: Quantum image processing. In: Proceedings of The 2003 IEEE Workshop on Applied Imagery Pattern Recognition, pp. 39–44 (2003)
21. Venegas-Andraca, S., Bose, S.: Quantum computation and image processing: new trends in artificial intelligence. In: Proceedings of the International Conference on Artificial Intelligence IJCAI-03, pp. 1563–1564 (2003)
22. Venegas-Andraca, S., Bose, S.: Storing, processing and retrieving an image using quantum mechanics. In: Proceedings of the SPIE Conference Quantum Information and Computation, pp. 137–147 (2003)
23. Zhang, W.W., Gao, F., Liu, B., Wen, Q.Y., Chen, H.: A watermark strategy for quantum images based on quantum Fourier transform. Quantum Inf. Process. **12**(2), 793 (2013)
24. Song, X.H., Wang, S., Liu, S., El-Latif, A.A.A., Niu, X.M.: A dynamic watermarking scheme for quantum images using quantum wavelet transform. Quantum Inf. Process. **12**(12), 3689 (2013)
25. Song, X., Wang, S., El-Latif, A.A.A., Niu, X.: Dynamic watermarking scheme for quantum images based on Hadamard transform. Multimed. Syst. **20**(4), 379 (2014)
26. Miyake, S., Nakamae, K.: A quantum watermarking scheme using simple and small-scale quantum circuits. Quantum Inf. Process. **15**(5), 1849–1864 (1849)
27. 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. **12**(11), 3477 (2013)
28. Song, X., Wang, S., El-Latif, A.A.A., Niu, X.: Quantum image encryption based on restricted geometric and color transformations. Quantum Inf. Process. **13**(8), 1765 (2014)
29. Zhou, N.R., Hua, T.X., Gong, L.H., Pei, D.J., Liao, Q.H.: Quantum image encryption based on generalized Arnold transform and double random-phase encoding. Quantum Inf. Process. **14**(4), 1193 (2015)
30. Gong, L.H., He, X.T., Cheng, S., Hua, T.X., Zhou, N.R.: Quantum image encryption algorithm based on quantum image XOR operations. Int. J. Theor. Phys. **55**(7), 3234 (2016)
31. Liang, H.R., Tao, X.Y., Zhou, N.R.: Quantum image encryption based on generalized affine transform and logistic map. Quantum Inf. Process. **15**(7), 2701 (2016)
32. Tan, R.C., Lei, T., Zhao, Q.M., Gong, L.H., Zhou, Z.H.: Quantum color image encryption algorithm based on a hyper-chaotic system and quantum fourier transform. Int. J. Theor. Phys. **10**, 5368–5384 (2016)
33. Zhou, N., Hu, Y., Gong, L., Li, G.: Quantum image encryption scheme with iterative generalized Arnold transforms and quantum image cycle shift operations. Quantum Inf. Process. **10**, 164 (2017)
34. El-Latif, A.A.A., Abd-El-Atty, B., Talha, M.: Robust encryption of quantum medical images. IEEE Access **6**, 1073–1081 (2017)
35. Li, L., Abd-El-Atty, B., El-Latif, A.A.A., Ghoneim, A.: Quantum color image encryption based on multiple discrete chaotic systems. In: Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 555–559 (2017)
36. Jiang, N., Zhao, N., Wang, L.: LSB based quantum image steganography algorithm. Int. J. Theor. Phys. **55**(1), 107 (2016)
37. Abd-El-Atty, B., El-Latif, A.A.A., Amin, M.: New quantum image steganography scheme with Hadamard transformation. In: International Conference on Advanced Intelligent Systems and Informatics, pp. 342–352. Springer (2016)
38. Zhang, T.J., Abd-El-Atty, B., Amin, M., El-Latif, A.A.A.: QISLSQb: a quantum image steganography scheme based on least significant qubit. In: DEStech Transactions on Computer Science and Engineering (MCSSE) (2016)
39. Wang, S., Sang, J., Song, X., Niu, X.: Least significant qubit (LSQb) information hiding algorithm for quantum image. Measurement **73**, 352 (2015)
40. El-Latif, A.A.A., Abd-El-Atty, B., Hossain, M.S., Rahman, M.A., Alamri, A., Gupta, B.: Efficient quantum information hiding for remote medical image sharing. IEEE Access **6**, 21075 (2018)

41. Zhang, Y., Lu, K., Gao, Y., Wang, M.: NEQR: a novel enhanced quantum representation of digital images. Quantum Inf. Process. **12**(12), 2833 (2013)
42. Le, P.Q., Dong, F., Hirota, K.: A flexible representation of quantum images for polynomial preparation, image compression and processing operations. Quantum Inf. Process. **10**(1), 63 (2011)
43. Childs, A.: Universal computation by quantum walk. Phys. Rev. Lett. **102**, 180501 (2009)
44. Venegas-Andraca, S.E.: Quantum walks: a comprehensive review. Quantum Inf. Process. **11**(5), 1015 (2012)
45. Shenvi, N., Kempe, J., Whaley, R.: A quantum random walk search algorithm. Phys. Rev. A **67**(5), 052307 (2003)
46. Childs, A., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., Spielman, D.: Exponential algorithmic speedup by quantum walk. In: Proceedings of the 35th ACM Symposium on The Theory of Computation (STOC'03), pp. 59–68. ACM (2003)
47. Santha, M.: Quantum walk based search algorithms. In: Proceedings of the 5th Theory and Applications of Models of Computation (TAMC08), Xian, LNCS 4978, pp. 31–46 (2008)
48. Childs, A., van Dam, W.: Quantum algorithms for algebraic problems. Rev. Mod. Phys. **82**, 1 (2010)
49. Yang, Y.G., Pan, Q.X., Sun, S.J., Xu, P.: Novel image encryption based on quantum walks. Sci. Rep. **5**(7784), 7784 (2015)
50. Yang, Y.G., Zhao, Q.Q.: Novel pseudo-random number generator based on quantum random walks. Sci. Rep. **6**, 20362 (2016)
51. Li, D., Zhang, J., Guo, F.Z., Huang, W., Wen, Q.Y., Chen, H.: Discrete-time interacting quantum walks and quantum Hash schemes. Quantum Inf. Process. **12**, 1–13 (2013)
52. Li, D., Yang, Y.G., Bi, J.L., Yuan, J.B., Xu, J.: Controlled alternate quantum walks based quantum hash function. Sci. Rep. **8**(1), 225 (2018)
53. Yang, Y.G., Xu, P., Yang, R., Zhou, Y.H., Shi, W.M.: Quantum Hash function and its application to privacy amplification in quantum key distribution, pseudo-random number generation and image encryption. Sci. Rep. **6**, 19788 (2016)
54. Aharonov, Y., Davidovich, L., Zagury, N.: Quantum random walks. Phys. Rev. A **48**, 1687 (1993)
55. Nayak, A., Vishwanath, A.: Quantum walk on the line. arXiv:quant-ph/0010117
56. Aharonov, D., Ambainis, A., Kempe, J., Vazirani, U.: Quantum walks on graphs. In: Proceedings of the 33th ACM Symposium on the Theory of Computation (STOC'01), pp. 50–59. ACM (2001)
57. Lovett, N., Cooper, S., Everitt, M., Trevers, M., Kendon, V.: Universal quantum computation using the discrete-time quantum walk. Phys. Rev. A **81**(4), 042330 (2010)
58. Feldman, E., Hillery, M.: Modifying quantum walks: a scattering theory approach. J. Phys. A Math. Theor. **40**, 11343 (2007)
59. Carson, G.R., Loke, T., Wang, J.B.: Entanglement dynamics of two-particle quantum walks. Quantum Inf. Process. **14**(9), 3193 (2015)
60. Luo, H., Xue, P.: Properties of long quantum walks in one and two dimensions. Quantum Inf. Process. **14**(12), 4361 (2015)
61. Wong, T.G.: Quantum walk on the line through potential barriers. Quantum Inf. Process. **15**(2), 675 (2016)
62. Konno, N., Mitsuhashi, H., Sato, I.: The discrete-time quarternionic quantum walk on a graph. Quantum Inf. Process. **15**(2), 651 (2016)
63. Tregenna, B., Flanagan, W., Maile, R., Kendon, V.: Controlling discrete quantum walks: coins and initial states. N. J. Phys. **5**(1), 83 (2003)
64. Gonzalez, R., Woods, R.: Digital Image Processing, 2nd edn. Prentice Hall, Upper Saddle River (1992)
65. Bitcoin Private Keys. https://wiki.bitcoin.com/w/Private_key. Accessed 1 June 2019