3D Res (2017) 8:4
DOI 10.1007/s13319-016-0114-7

3DR EXPRESS

A Novel Color Image Encryption Algorithm Based on Quantum Chaos Sequence

Hui Liu · Cong Jin

Received: 5 September 2016 / Revised: 16 December 2016 / Accepted: 19 December 2016 / Published online: 30 January 2017

Abstract In this paper, a novel algorithm of image encryption based on quantum chaotic is proposed. The keystreams are generated by the two-dimensional logistic map as initial conditions and parameters. And then general Arnold scrambling algorithm with keys is exploited to permute the pixels of color components. In diffusion process, a novel encryption algorithm, folding algorithm, is proposed to modify the value of diffused pixels. In order to get the high randomness and complexity, the two-dimensional logistic map and quantum chaotic map are coupled with nearest-neighboring coupled-map lattices. Theoretical analyses and computer simulations confirm that the proposed algorithm has high level of security.

Keywords Image encryption · Arnold scrambling · Folding algorithm · Quantum chaotic map · Two-dimensional logistic map

1 Introduction

1.1 Background

Currently, the image encryption technology is a hot area and a challenging task. There are lots of image information received by illegal users, which brings many negative impacts on personal privacy. In order to protect personal information, various image encryption algorithms are designed and proposed such as one-time keys [1], bit-level permutation [2, 3], compression techniques [4], DNA computing [5–7], Arnold transform [8–13] and so on. Chaotic systems have many good characteristics such as sensitivity to initial parameters, mixing property, high efficiency and ergodicity. In general, a chaotic system has a high speed with low cost, which is better than many conditional ciphers for multimedia data encryption [14]. Inspired by the subtle similarity between chaotic systems and cryptosystem, various encryption algorithms based on chaotic map [15, 16] are proposed in the literature. Herein, quantum chaotic system is applied to generate pseudo-random sequence to encrypt color images in the proposed cryptosystem.

An international standard of encryption algorithm is not only suitable for a partial compression algorithm but also permutation and diffusion properties. The diffusion–permutation-based algorithm should have a large key space and the long periodicity of permutation to increase the security. For this purpose, many researchers turn to find some improved chaos-based algorithms with large key spaces and good permutation and diffusion techniques. Ping [1] proposed a novel CA-based multiple image encryption by using a kind of two-dimensional reversible CA, and by using a circular chaining mode of operation. The proposed method allows images to be processed in a 2-D way

H. Liu · C. Jin (✉)
School of Computer, Central China Normal University, Wuhan 430079, Hubei, China
e-mail: jincong@mail.ccnu.edu.cn

and makes the statistical information of each plain image in the group hidden in all cipher images.

In order to disturb the high correlation among pixels, the Arnold cat map [8–13] is a good scrambling tool which has been used widely in various cryptographic and steganographic applications. Guo et al. [9] proposed a novel color image encryption method using discrete fractional random transform (DFRNT) and Arnold transform (AT) in the intensity–hue–saturation (IHS) color space. Chen et al. [12] reported a new image encryption algorithm based on singular value decomposition and Arnold transform. However, in all of these algorithms have a weaknesses [17]: the iteration times are very limited. Here we propose perfect methods to solve these problems.

Chaos-based cryptographic scheme has many brilliant advantages different from other algorithms such as sensitivity to initial conditions and parameters, mixing property, high efficiency non-periodicity and control parameters [18, 19]. In recent years various encryption algorithms based on chaotic map are proposed [20–22]. Wang and Guo [20] utilized a logistic map for generating a matrix to diffuse the left block of the plain image and then the diffused image was used as the right block of the cipher image. In [23] quantum chaos theory becomes a tool that can be used to improve the quality of pseudo-random number generators. The randomness and non-periodicity of quantum chaotic map are successfully verified by statistical complexity and the normalized Shannon entropy. In order to obtain high diffusivity, folding algorithm is proposed to modify the value of permuted pixels with quantum chaos sequence from eight directions.

1.2 Contribution and Organization

Due to the color image that is composed of three color components, we convert three components into three matrices, namely R, G, B. General Arnold transform with keys means that parameters of the matrix A is a set of secret values. We add the matrix $(ku, kv)^T$ as secret values during the process that Arnold transform is iterated n times. The experiment proves that the chaos character is better when $n = 6$. So we get three different matrices $(ku_i, kv_i)^T$ $(i = 1, 2, 3)$ as keys to improve the high randomness and enlarge the key space. And then quantum chaotic map [24–26] is applied to generate three matrices X, Y, Z of size $N \times N$ to encrypt three matrices R, G and B. In this process, the initial condition of quantum chaotic map is a pseudo-random number, which is altered with the time of iteration. For the high complexity and the high randomness, in this paper chaotic maps are coupled with nearest-neighboring coupled-map (NCML), which extremely increases the security and sensitivity of the proposed algorithm.

The major contributions of the proposed algorithm are as follows:

(1) Add matrices $(ku_i, kv_i)^{\mathrm{T}}$ (i = 1, 2, 3) as keys into general Arnold transform to enlarge the key space and improve the randomness.
(2) Key generator is an address mapping table, which is generated by two-dimensional logistic map. According to session keys we obtain initial conditions and parameters so that improve the sensitivity of the key generator.
(3) Putting forward a new algorithm, the folding algorithm, to encrypt the image from eight directions and attaining remarkable results.

The rest of this paper is organized in the following manners: Sect. 2 introduce the basic theory of the proposed cryptosystem. Section 3 the proposed cryptosystem is explained. Simulation results and security analysis are proposed in Sect. 4. Finally the conclusions are drawn in Sect. 5.

2 Basic Theory of the Cryptosystem

2.1 Two-Dimensional Logistic Map

As a classical algorithm logistic has a perfect chaotic property. The 2D coupled logistic map reported in [27] has three quadratic coupling terms to strengthen its complexity. In order to enlarge key space and obtain the high complexity, in this paper two-dimensional logistic map is chosen to be a key generator. The two-dimensional logistic map is described as [27, 28]:

$$\begin{aligned} \varphi_1(x_n) &= \mu_1 x_n(1 - x_n) + \gamma_1 y_n^2 \\ \varphi_1(y_n) &= \mu_2 y_n(1 - y_n) + \gamma_2(x_n^2 + x_n y_n), \end{aligned} \tag{1}$$

when $2.75 < \mu_1 \leq 3.4$, $2.75 < \mu_2 \leq 3.45$, $0.15 < \gamma_1 \leq 0.21$ and $0.13 < \gamma_2 \leq 0.15$, the system can generate pseudo-numbers in the region (0,1]. All parameters are generated by key generator.

2.2 General Arnold Transform with Keys

In order to disturb the high correlation among pixels, the Arnold cat map [8] is image pixel scrambling tool which has been used in many literatures. The definition of general Arnold transform is given in [29]:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} (\mathrm{mod}\ N), \quad A = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix}, \tag{2}$$

where the location of the plain-image pixel is (x, y), the location of the cipher-image pixel is (x′, y′). We set N = 256. When a = b = 1, Eq. (2) is a classical two-dimensional Arnold map. In order to improve security of the cryptosystem, the control parameters a and b are generated by the key generator. Because Arnold transform is a bijection transform, the result of iterating Eq. (2) k times still is a bijection transform. In other words, after the process of iteration for k times, point (x′, y′) is new position of coordinate (x, y). Due to the fact that result of orthogonal transformation is a limited discrete set, we can add a matrix $(ku, kv)^{\mathrm{T}}$ as a set of secret keys to reduce the correlation among pixels. So we get general Arnold transform with keys as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A^n \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} ku \\ kv \end{bmatrix} (\mathrm{mod}\ N), \quad A = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix}, \tag{3}$$

where n is iteration times of the matrix A. According to the inverse transformation of Eq. (3), the corresponding decryption algorithm is shown as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = A^{-n} \begin{bmatrix} x' - ku \\ y' - kv \end{bmatrix} (\mathrm{mod}\ N), \quad A^{-1} = \begin{bmatrix} ab+1 & -a \\ -b & 1 \end{bmatrix}. \tag{4}$$

2.3 Quantum Chaotic Map

Dissipative quantum systems are often described in where the system is coupled to a path of harmonic oscillators to construct a quantum logistic map [24–26] with quantum corrections. In [24], authors analyze the effects of quantum corrections and state $\alpha = \langle\alpha\rangle + \delta\alpha$, where $\delta\alpha$ shows a quantum fluctuation about $\langle\alpha\rangle$. Furthermore, they prove that the very lowest-order quantum corrections can yield the chaotic map as follows:

$$\begin{aligned} \varphi_2(x'_n) &= r\,(x'_n - |x'_n|^2) - r\,y'_n \\ \varphi_2(y'_n) &= -y'_n e^{-2\beta} + e^{-\beta} r\,[(2 - x'_n - x'^{*}_n) y'_n - x'_n z'^{*}_n - x'^{*}_n z'_n] \\ \varphi_2(z'_n) &= -z'_n e^{-2\beta} + e^{-\beta} r\,[2(1 - x'^{*}_n)\, z'_n - 2x'_n y'_n - x'_n] \end{aligned} \tag{5}$$

where $\acute{x} = \langle\alpha\rangle$, $\acute{y} = \langle\delta\alpha^{\dagger}\ \delta\alpha\rangle$, $\acute{z} = \langle\delta\alpha\ \delta\alpha\rangle$ and β is dissipation parameter. Generally, x'_n, y'_n and z'_n are complex numbers with x'^{*}_n being the complex conjugate of x'_n and similarly for z'_n. However, if we set the initial value to be real number, then all successive value will also be real. According to [23], the range of the parameters as follows: $0 \le x'_n \le 1$, $0 \le y'_n \le 0.1$, $0 \le z'_n \le 0.2$, $x'^{*}_n = x'_n$, $z'^{*}_n = z'_n$, $\beta \in [6, +\infty]$ and $r \in [0, 4]$. They conclude that the best value of the control parameter r and dissipation parameter β are r = 3.99 and β ≥ 6. So we set r = 3.99, β = 6. Iterate Eq. (5) with real initial parameters x'_0, y'_0, z'_0, x'^{*}_0 and z'^{*}_0, all the successive values x'_n, y'_n and z'_n will be real.

2.4 Nearest-Neighboring Coupled-Map Lattices

In order to achieve the high complexity and the high randomness among these generated keystreams, the two-dimensional logistic map and the quantum chaotic map proposed in Sects. 2.1 and 2.3 are independently coupled with NCML [30, 31] as follows:

$$z_{n+1}(j) = (1 - \varepsilon)\varphi(z_n(j)) + \varepsilon\varphi(z_n(j+1)), \tag{6}$$

where n = 0, 1,…, L − 1 is the time index; j = 1, 2,…, T is the lattice state index; function φ represents a chaotic map such as φ_1, φ_2; $\varepsilon \in (0, 1)$ is a coupling constant; L is the length of the plain-text; and T is maximum value of lattice state index. Here, T is chosen as 2 and 3 for the two-dimensional logistic map and the quantum chaotic map, while the other parameter is selected as $\varepsilon = 0.001$ to have good chaotic properties [30, 31]. Moreover, the periodic boundary condition, i.e., $z_n(j + T) = z_n(j)$ is imposed into this system.

Applying Eqs. (1) to (6), the coupling of two-dimensional logistic map is defined as follows:

$$\begin{aligned} x_{n+1} &= (1 - \varepsilon)\varphi(x_n) + \varepsilon\varphi(y_n) \\ y_{n+1} &= (1 - \varepsilon)\varphi(y_n) + \varepsilon\varphi(x_n) \end{aligned} \tag{7}$$

and by applying Eqs. (5) to (6), the coupling of quantum chaotic map is defined as follows:

$$\begin{aligned} x'_{n+1} &= (1-\varepsilon)\varphi(x'_n) + \varphi(y'_n) \\ y'_{n+1} &= (1-\varepsilon)\varphi(y'_n) + \varphi(z'_n) \\ z'_{n+1} &= (1-\varepsilon)\varphi(z'_n) + \varphi(x'_n) \end{aligned} \tag{8}$$

Iterating Eqs. (7) and (8), the required keystreams for the proposed cryptosystem are produced.

3 Cryptosystem

In what follows, we combine the generation process with the image processing, the permutation process and the diffusion process.

3.1 Generation of the Initial Conditions and Parameters

Proposed cryptosystem utilizes a 128-bit external secret key, K, which is divided into 8-bit blocks, k_i, referred to as session keys. The 128-bit external secret key is given by:

$$K = k_1, k_2, \ldots, k_{16} \tag{9}$$

In order to increase the security of the proposed algorithm, we apply the two-dimensional logistic map Eq. (1) and nearest-neighboring coupled-map lattices Eq. (6) so that the initial conditions and parameters of the system are extremely sensitive to the changes in even a single bit in the 128-bit secret key. The detailed process of key generator is described as follows:

Step 1 Apply k_1, k_2, k_3, k_4 to generate μ_1, μ_2, γ_1, γ_2 respectively. We have known that when $2.75 < \mu_1 \leq 3.4$, $2.75 < \mu_2 \leq 3.45$, $0.15 < \gamma_1 \leq 0.21$ and $0.13 < \gamma_2 \leq 0.15$ the two-dimensional logistic map generates chaos. We set $a < t_i \leq b$, the initial conditions and parameters of system are derived as follows:

$$t_i = \left\{ \left(\frac{k_i}{256} \times 100 \right) mod[(b-a) \times 100] \right\} / 100 + a, \tag{10}$$

where we set $\mu_1 = t_1$, $\mu_2 = t_2$, $\gamma_1 = t_3$, $\gamma_2 = t_4$. So for the different k_i we can get different t_i and make sure that μ_1, μ_2, γ_1, γ_2 are in the region that the system generate chaos.

Step 2 Apply k_5, k_6,…, k_{16} as initial condition to generate other key values. $t_{max} = \max([k_5, k_6, \ldots, k_{16}])$. $t_{min} = \min([k_5, k_6, \ldots, k_{16}])$. $t_{ssv} = \min([k_5, k_6, \ldots, k_{16}] - t_{min})$. We set $x_0 = t_{min}/256$, $y_0 = t_{ssv}/256$ and iterate Eq. (7) for ceil ($t_{max}/2$) times with μ_1, μ_2, γ_1, γ_2, x_0, y_0 and then save their output in a new vector E whose size is $2 \times$ ceil ($t_{max}/2$). Apply the following Eq. (11):

$$t_i = E_{k_i} \tag{11}$$

where $i = 5, 6, \ldots, 16$ and t_i are in the region (0, 1].

Step 3 In order to improve randomness and complexity of the encryption algorithm and broaden the key space, According to Eq. (4) three sets of secret keys, a_i, b_i and $(ku_i, kv_i)^T$, are required to encrypt three component of the color image R, G, B respectively. Without loss of generality, we assume that the size of the color plain-image P is $W \times H$. Apply the transformation as following equations to t_5, t_6, t_7:

$$\begin{aligned} a_{i-4} &= [\text{floor}(t_i \times W \times H) \bmod 256]/16 \\ b_{i-4} &= [\text{floor}(t_i \times W \times H) \bmod 256] \bmod 16 \end{aligned} \tag{12}$$

where a_i, b_i ($i = 1, 2, 3$) are the first four digits and the last four digits of eight-digit binary number respectively.

Apply the transformation as following equations to t_8, t_9, t_{10}:

$$ku_{i-7} = \text{floor}(t_i \times W \times H) \bmod 256. \tag{13}$$

Apply the transformation as following equations to t_{11}, t_{12}, t_{13}:

$$kv_{i-10} = \text{floor}(t_i \times W \times H) \bmod 256. \tag{14}$$

Step 4 Recalling as mention in Sects. 2.3, $y'_n \in [0, 0.1]$, $z'_n \in [0, 0.2]$. Applying Eq. (10) analogously initial parameters x'_0, y'_0, z'_0 are derived as follows:

$$\begin{aligned} x'_0 &= t_{14} \\ y'_0 &= [(t_{15} \times 10) \bmod 1]/10 \\ z'_0 &= [(t_{16} \times 10) \bmod 2]/10 \end{aligned} \tag{15}$$

To this end, all initial conditions and parameters are generated. The above key generator shows that we can not find different keys which make the same effect on initial parameters. And the proposed chaotic algorithm is greatly sensitive to secret key so that even a change in the secret key causes completely different results; as a result, the proposed algorithm with total complexity of 2^{128} can resist against any key sensitivity attack and any bruteforce attack.

3.2 Encryption Algorithm

In this process we convert the matrix P with red, green and blue components into three matrices R, G and B. Taking an example of the matrix R, the detailed encryption algorithm is described as follows:

3.2.1 Permutation Process

The process applies pseudo-random keystreams generated by Eqs. (12), (13) and (14) according to Sect. 3.1 to permute pixels of the color image. Substituting a_1, b_1 and $(ku_1, kv_1)^T$ into Eq. (3) and iterate it for n times. According to the experiment we find that when $n = 6$ the proposed cryptosystem performs better. Apply the same permutation process into G and B respectively, the plain-image becomes a cipher-image after n times iteration, namely, Matrices R, G and B all becomes R', G' and B'.

3.2.2 Diffusion Process

Step 1 Set $L = N \times N$ and generate the initial condition (x'_0, y'_0, z'_0) according to Sect. 3.1 and iterate Eq. (8) $m + L$ times and discard the former m values to avoid harmful effects. Where m also can be as a secret key, we set $m = 13$ for convenience. Discarding the first m result and Sorting these L values as $|X> = \{x_{m+1}, x_{m+2},\ldots, x_{m+L}\}$, $|Y> = \{y_{m+1}, y_{m+2},\ldots, y_{m+L}\}$ and $|Z> = \{z_{m+1}, z_{m+2},\ldots, z_{m+L}\}$.

Step 2 Transforming three vectors |X>, |Y> and |Z> into matrices X, Y and Z respectively, whose size are $N \times N$.

Step 3 In order to describe the problem clearly, Red channel is used to be an example to explain "the process of folding the picture". We fold the matrix R (Fig. 2a) from eight directions to encrypt it. Eight directions include eight rounds encryption ways.

Round 1 (Fig. 1):

$$\begin{aligned} Th'(i,j) &= Th(i,j) \oplus Xth(i,j) \\ Bh'(N-i+1,j) &= Bh(N-i+1,j) \oplus Th'(i,j) \end{aligned} \tag{16}$$

where $i = 1, 2,\ldots, N/2$ and $j = 1, 2,\ldots, N$. The matrix R' is divided into two equal horizontal parts: Th and Bh.

Round 2 (Fig. 2):

$$\begin{aligned} Tr'(i,j) &= Tr(i,j) \oplus Xtr(i,j) \\ Bl'(j,i) &= Bl(j,i) \oplus Tr'(i,j) \end{aligned} \tag{17}$$

where $i = 1, 2,\ldots, N$ and $j = i, i+1,\ldots, N$. The matrix R1 is divided into two equal horizontal parts: Tr and Bl.

Round 3 (Fig. 3):

$$\begin{aligned} Rh'(i,j) &= Rh(i,j) \oplus Xrh(i,j) \\ Lh'(i,N-j+1) &= Lh(i,j) \oplus Rh'(i,N-j+1) \end{aligned} \tag{18}$$

where $i = 1, 2,\ldots, N$ and $j = N/2 + 1, N/2 + 2,\ldots, N$. The matrix R2 is divided into two equal horizontal parts: Lh and Rh.

Round 4 (Fig. 4):

$$\begin{aligned} Rb'(i,j) &= Rb(i,j) \oplus Xrb(i,j) \\ Lt'(i,j) &= Rb'(i,j) \oplus Lt(i,j) \end{aligned} \tag{19}$$

where $i = 1, 2,\ldots, N$ and $j = N - i + 1, N - i + 2,\ldots, N$. The matrix R3 is divided into two equal horizontal parts: Lt and Rb.

Round 5 (Fig. 5):

$$\begin{aligned} Bh'(i,j) &= Bh(i,j) \oplus Xbh(i,j) \\ Th'(N-i+1,j) &= Th(N-i+1,j) \oplus Bh'(i,j) \end{aligned} \tag{20}$$

where $i = N/2 + 1, N/2 + 2,\ldots, N$ and $j = 1, 2,\ldots, N$. The matrix R4 is divided into two equal horizontal parts: Th and Bh.

Round 6 (Fig. 6):

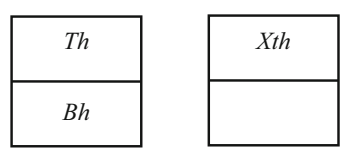
$$\begin{aligned} Bl'(i,j) &= Bl(i,j) \oplus Xbl(i,j) \\ Tr'(j,i) &= Tr(j,i) \oplus Bl'(i,j) \end{aligned} \tag{21}$$

where $i = 1, 2,\ldots, N$ and $j = 1, 2,\ldots, i–1$. The matrix R5 is divided into two equal horizontal parts: Tr and Bl.

Round 7 (Fig. 7):

$$\begin{aligned} Lh'(i,j) &= Lh(i,j) \oplus Xlh(i,j) \\ Rh'(i,N-j+1) &= Rh(i,j) \oplus Lh'(i,N-j+1) \end{aligned} \tag{22}$$

where $i = 1, 2,\ldots, N$ and $j = 1, 2,\ldots, N/2$. The matrix R6 is divided into two equal horizontal parts: Lh and Rh.

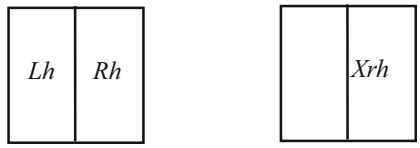


(a) The matrix *R'* **(b)** The matrix *X*

Fig. 1 Fold from top to bottom. a The matrix R'. b The matrix X

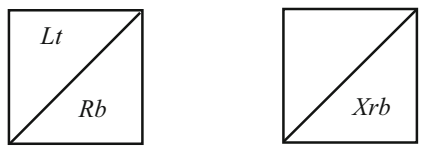
(a) The matrix *R*1 after round 1 **(b)** The matrix *X*

Fig. 2 Fold from top right to bottom left. a The matrix R1 after round 1. b The matrix X



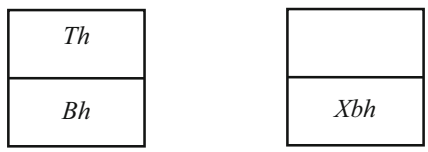
(a) The matrix *R*2 after round 2 **(b)** The matrix *X*

Fig. 3 Fold from right to left. a The matrix R2 after round 2. b The matrix X



(a) The matrix *R3* after round 3 **(b)** The matrix *X*

Fig. 4 Fold from bottom right to top left. a The matrix R3 after round 3. b The matrix X



(a) The matrix *R*4 after round 4 **(b)** The matrix *X*

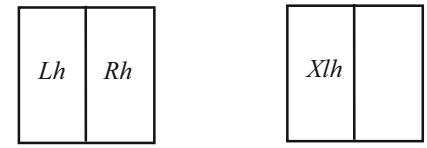
Fig. 5 Fold from bottom to top. a The matrix R4 after round 4. b The matrix X

(a) The matrix *R*5 after round 5 **(b)** The matrix *X*

Fig. 6 Fold from bottom left to top right. a The matrix R5 after round 5. b The matrix X

Round 8 (Fig. 8):

$$\begin{aligned} Lt'(i,j) &= Lt(i,j) \oplus Xlt(i,j) \\ Rb'(i,j) &= Rb(i,j) \oplus Lt'(i,j) \end{aligned} \tag{23}$$



(a) The matrix *R*6 after round 6 **(b)** The matrix *X*

Fig. 7 Fold from left to right. a The matrix R6 after round 6. b The matrix X

where i = 1, 2,…, N and j = 1, 2,…, N–i. The matrix R7 is divided into two equal horizontal parts: Lt and Rb.

Step 4 After Step 3 the matrix R′ becomes a matrix R8. At last R8 XOR X and we get the encrypted matrix C_r.

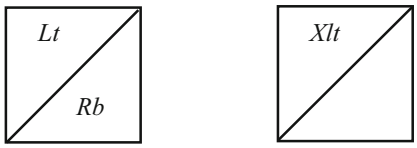
After four steps, the matrix R′ becomes an encrypted matrix C_r. The process of R component encryption is finished. In a similar way, we replace the matrix X with Y or Z and replace the matrix R′ with G′ or B′ respectively. After four steps the matrices G′ and B′ encrypted matrices C_g and C_b.

It is noted that M mod N involves modulo operation giving an integer result between 0 and N. Function ceil (a) returns the smallest integer value that is bigger than or equal to the value of a. Function max (k_1, k_2,…, k_n) returns the biggest value among all of them. And function min (k_1, k_2,…, k_n) returns the smallest value among all of them.

Obviously the generation of the keystream depends on the 128-bit external secret key, K, and the size N of plain-image. The generation of initial conditions and parameters are derived by the two-dimensional logistic map and the nearest-neighboring coupled-map lattices. And the keystream is chosen from an array of chaotic sequence, which makes sure that cryptosystem has a high complexity, sensitivity and randomness. In the encryption process, the Arnold transform with keys is applied to permute the pixels of color components. And the quantum chaotic map is exploited to generate the keystreams to modify the value of diffused pixels by "the process of folding the picture".

3.3 Decryption Algorithm

The decryption process is similar to the encryption one, achieved in the reverse order. In decryption process opening folded matrices C_r, C_g and C_b is the first steps. Second by applying the Eq. (4) we can accomplish encryption of Arnold transform. The detail decryption algorithm is described as follows:



(a) The matrix $R7$ after round 7 **(b)** The matrix X

Fig. 8 Fold from top left to bottom right. a The matrix R7 after round 7. b The matrix X

Step 1 According to Sect. 3.1 applying the same external 128-bit secret key to generate the initial conditions and parameters.

Step 2 Substituting the initial condition (x'_0, y'_0, z'_0) and iterating Eq. (8) m + L times, discarding the former m values to avoid harmful effects, where m = 13.

Step 3 Sorting these values $\{x_{m+1}, x_{m+2},\ldots, x_{m+L}\}$, $\{y_{m+1}, y_{m+2},\ldots, y_{m+L}\}$ and $\{z_{m+1}, z_{m+2},\ldots, z_{m+L}\}$ and transforming them to three matrices X, Y and Z.

Step 4 Executing these operations of C_r XOR X, C_g XOR Y and C_b XOR Z.

Step 5 In order to describe the process of decryption clearly, Round 1 is taken an example to explain the process of "opening folded matrices".

$$\begin{aligned} Dh(N-i+1,j) &= Dh'(N-i+1,j) \oplus Uh'(i,j) \\ Uh(i,j) &= Uh'(i,j) \oplus Xuh(i,j) \end{aligned} \tag{24}$$

where i = 1, 2,…, N/2 and j = 1, 2,…, N. Uh′, Dh′ and Xuh are all known quantity. As the same way, after eight rounds the process of "opening folded matrices" finished and we get three matrices R_r, G_g and B_b whose size are all N × N.

Step 6 Substituting the initial condition $(ku_i, kv_i)^T$ and parameters a_i, b_i (i = 1, 2, 3), and then using the encryption algorithm Eq. (4) we get R, G and B. In this way the encryption process finished.

4 Performance and security analysis

A good encryption algorithm should resist all kinds of known attacks, such as exhaustive attack, statistical attack and chosen-plaintext/ciphertext attack [32]. We have done many measures to check the security and performance of the proposed cryptosystem. These measures consist of statistical analysis, key sensitivity analysis, key space analysis, speed performance. Each of these measures is shown in detail in the following subsections.

4.1 Statistical Analysis

4.1.1 Histogram of Encrypted Image

An ideal cipher-image should have a uniform frequency distribution. From Figs. 9, 10, 11 and 12, it is obvious that the histogram of cipher-image are independent of the type of plain-image such as binary, gray level and are nearly uniform and significantly different from the histogram of the original images. Hence it dose not provide any useful statistic data in the cipher-image to trigger any statistical attacks to the algorithm.

For quantity analysis for each keys, variances of histograms is employed to evaluate the uniformity of distributions of pixels. The lower value of variances indicate the higher uniformity of cipher-image. The variances of histograms is presented as follows [33]:

$$\mathrm{var}(Z) = \frac{1}{256^2}\sum_{i=1}^{256}\sum_{j=1}^{256}\frac{1}{2}(z_i - z_j)^2, \tag{25}$$

where Z is the vector of the histogram values and $Z = \{z_1, z_2,\ldots, z_{256}\}$. z_i and z_j are the numbers of pixels which values are equal to i and j respectively. According to [31] the variance value is 625571.4908 for histogram of the plain-image Lena. And in the paper the variance value is 5258.7134 for histogram of the cipher-image Lena. Therefore, the proposed algorithm is efficient.

4.1.2 Correlation of Two Adjacent Pixels

In order to get the correlation of two adjacent pixels we have selected 3000 pairs of two adjacent pixels from plain-image and cipher-image randomly for the experiment and have calculated the correlation coefficients as follows:

$$\begin{aligned} E &= \frac{1}{N}\sum_{i=1}^{N} x_i \\ D(x) &= \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \\ Cov(x,y) &= \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \\ r_{xy} &= \frac{Cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \end{aligned} \tag{26}$$

The x, y represents gray-level values of two adjacent pixels. The distribution of two horizontally adjacent

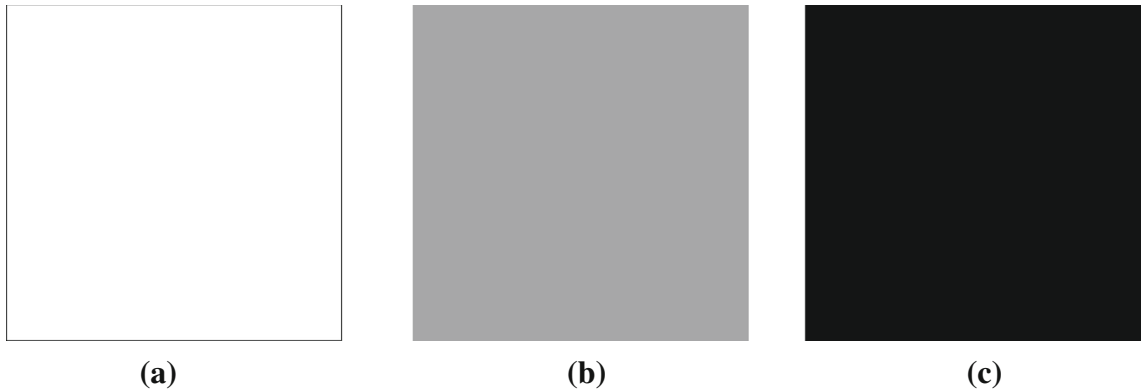


Fig. 9 a The original white image. b The original monolithic gray-level image. c The original black image

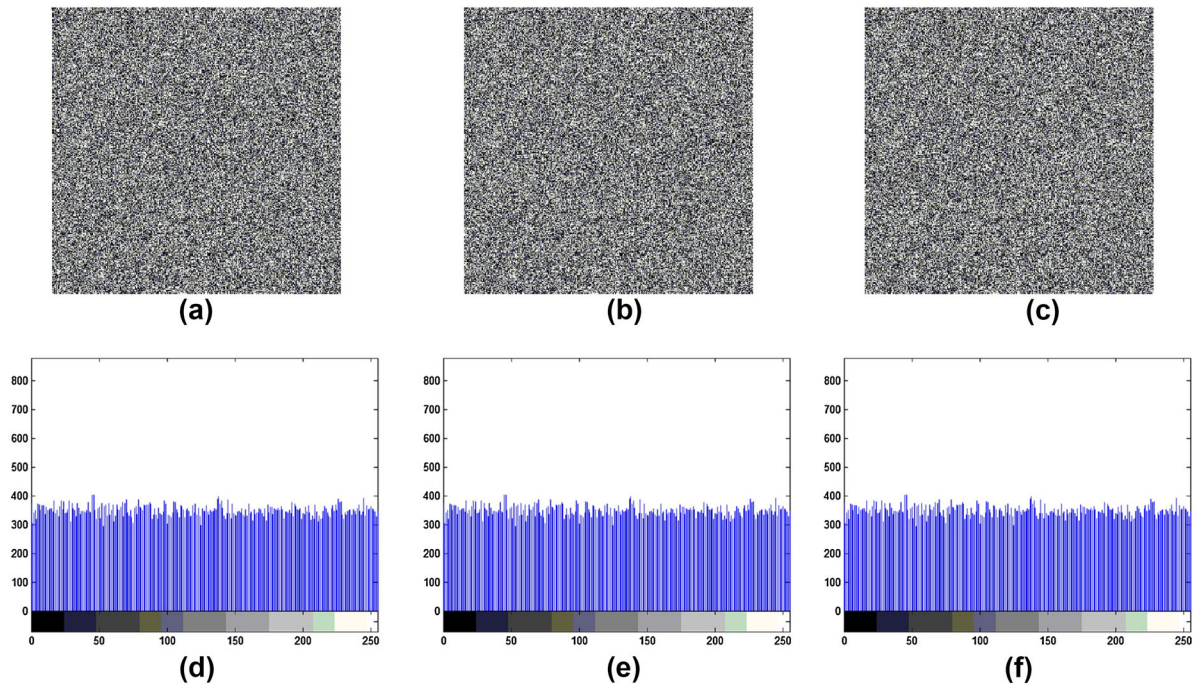


Fig. 10 a Cipher of white image. b The cipher of monolithic gray-level image. c The cipher of the black image. d The histogram of the encrypted white image. e The histogram of the encrypted monolithic gray-level image. f The histogram of the encrypted black image

pixels of R, G and B components of plain-image and cipher-image "Lena" is shown in Figs. 6 and 13.

Table 1 shows that the correlation between adjacent pixels of the cipher-image is much smaller than that of plain-image, so we claim that the adjacent pixels of the plain-image are uncorrelated by the proposed cryptosystem effectively from different directions.

In color images, there are the high correlations between adjacent pixels of R, G and B components. The proposed cryptosystem encrypt pixels of color components so that make them affect one another. Tables 2 and 3 show the results of the same position correlations and related adjacent position correlations between R, G and B components of plain-image and cipher-image.

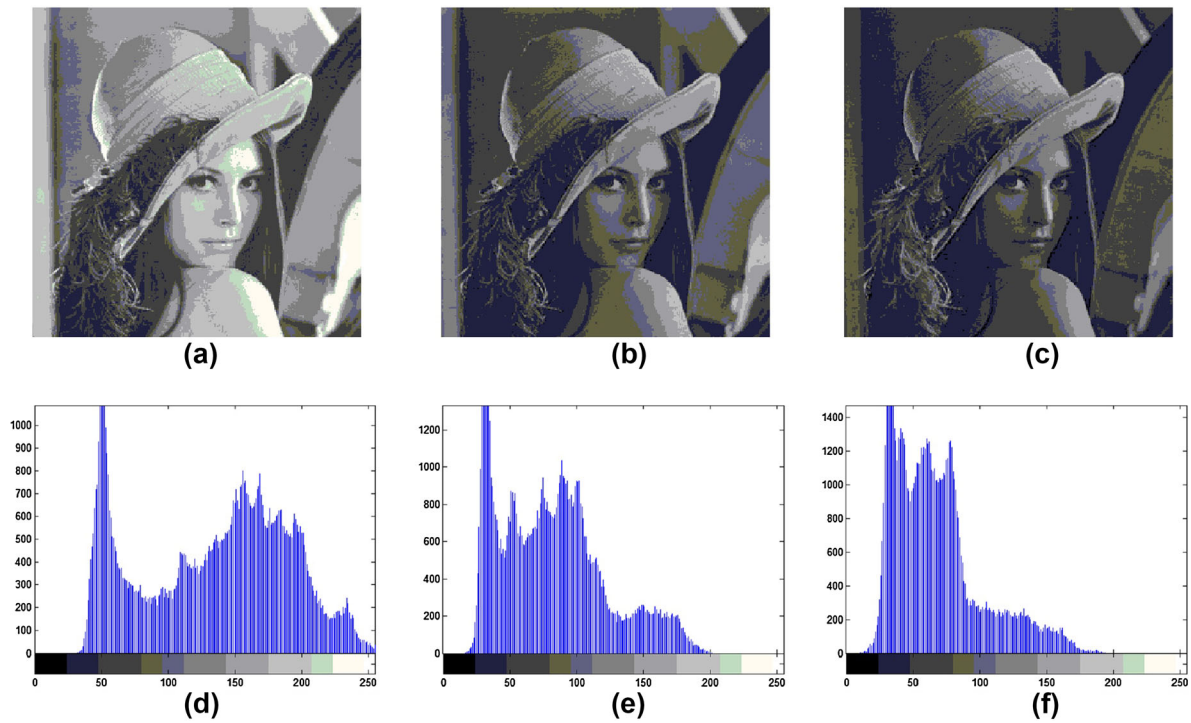


Fig. 11 a Plain-image Lena-R. b The plain-image Lena-G. c The plain-image Lena-B. d The histogram of the plain-image Lena-R. e The histogram of the plain-image Lena-G. f The histogram of the plain-image Lena-B

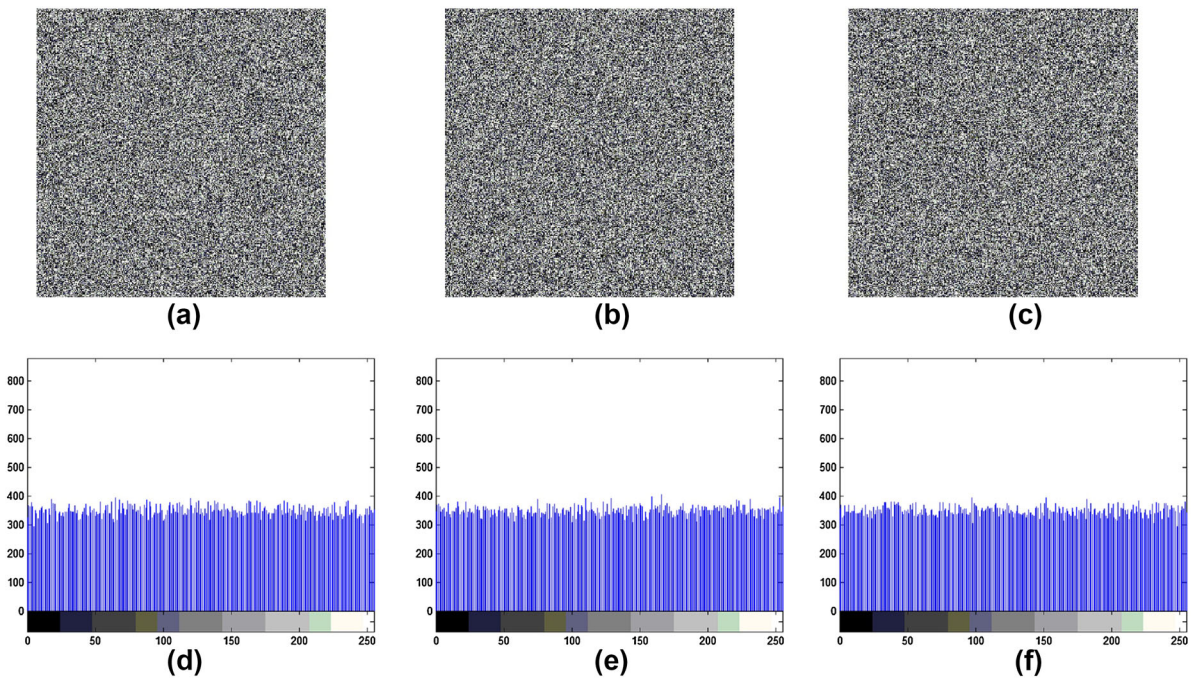


Fig. 12 a The encrypted image Lena-R. b The encrypted image Lena-G. c The encrypted image Lena-B. d The histogram of the encrypted image Lena-R. e The histogram of the encrypted image Lena-G. f The histogram of the encrypted image Lena-B

4.2 Key Sensitivity Analysis

When one bit of the security key is altered, there are obviously differences between two cipher-images. The number of pixels change rate (NPCR) and the unified average changing intensity (UACI) for the two encrypted images are applied to measure the number of pixels change rate.

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

$$\mathrm{UACI} = \frac{1}{N \times N}\left[\sum_{i,j} \frac{|C(i,j) - C(i,j)'|}{255}\right] \times 100\%, \tag{27}$$

where N is the height (width) of the encrypted image. We apply two encrypted images C and C′, whose corresponding original images are different in only one pixel. We also define a two-dimensional array D, which has the same size as C and C′. If C(i, j) = C′(i, j), then D(i, j) = 0, otherwise D(i, j) = 1. To resist against security key attack, NPCR and UACI values should be large enough for an ideal cipher system.

When the secret key is altered from "207 21 42 61 122 203 97 76 101 5 7 241 139 28 98 17" to "208 21 42 61 122 203 97 76 101 5 7 241 139 28 98 17" the differences is made greatly. Table 4 shows the average $\mathrm{NPCR}_{R,G,B}$ and $\mathrm{UACI}_{R,G,B}$ values and compares this proposed algorithm with other schemes in terms of the key sensitivity. The proposed algorithm is sensitive dependent on initial conditions and parameters and is effective to resist differential attack.

4.3 Key Space Analysis

An ideal encryption scheme should have an enough large key space to defend brute-force attack. The size of the key space should be bigger than 2^{100} to provide a high level of security from the crytography of view [38, 39]. Due to the secret key is 128-bit long, the key space is 2^{128}. We can conclude that the proposed algorithm is large enough to resist all kinds of brute-force attacks.

4.4 Speed Performance

Apart from the security considerations, some other aspects on image cryptosystem algorithm are also

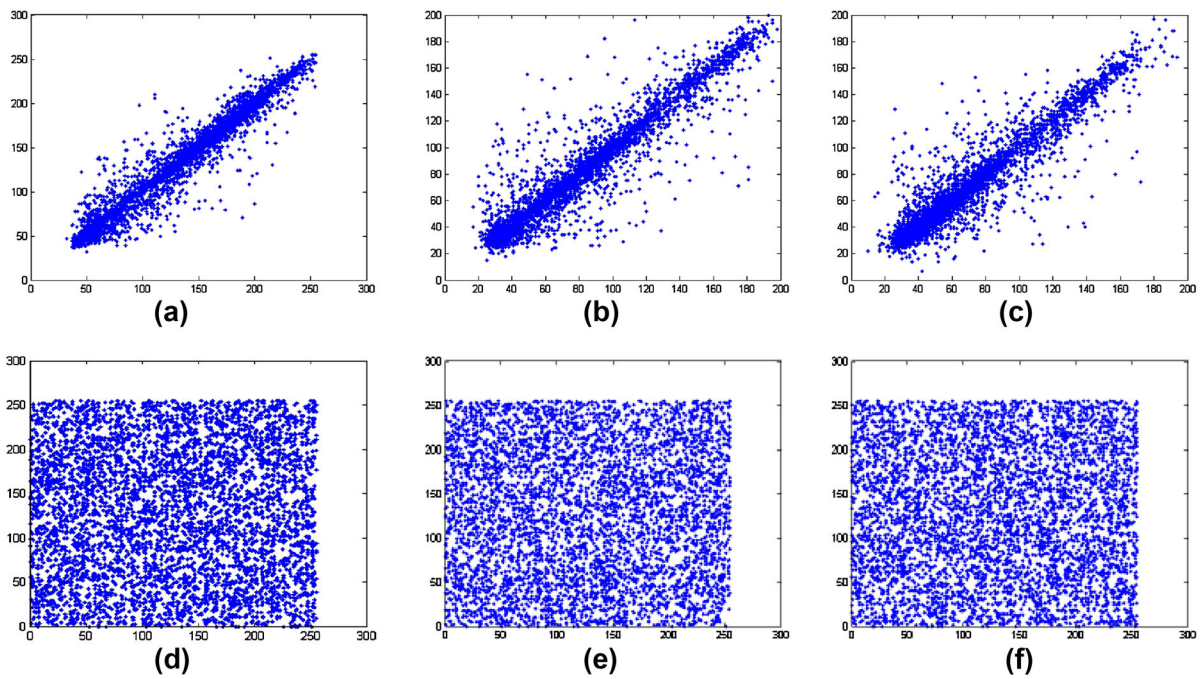


Fig. 13 Distribution of two horizontally adjacent pixels in the plain-image of Lena, (**a**) in the red. (**b**) in the green. (**c**) in the blue components. The distribution of two horizontally adjacent pixels in the cipher-image of Lena, (**d**) in the red, (**e**) in the green, (**f**) in the blue components. (Color figure online)

Table 1 The related correlation coefficient between plain-image and cipher-image

Scan direction	Lena					
	Plain-image			Cipher-image		
	R	G	B	R	G	B
Horizontal	0.972978	0.954127	0.938846	0.000215	0.000101	0.000147
Vertical	0.981110	0.951084	0.934597	0.000239	0.000563	0.000086
Diagonal	0.958757	0.934720	0.915541	0.000540	0.000632	0.001526

Table 2 Similar position correlations between R, G and B components

Scan direction	R–G	R–B	G–B
Plain-image	0.929848	0.797885	0.949200
Cipher-image	0.000235	0.001223	0.003218

Table 3 Adjacent position correlation between R, G and B components

Scan direction	R–G	R–B	G–B
Plain-image	0.896510	0.756614	0.891265
Cipher-image	0.002648	0.004300	0.001657

important, particularly the running speed for real time Internet multimedia applications. In fact the actual execution time of a cryptosystem depends on many factors, such as CPU structure, OS, memory size, programming skill and so on. We have analyzed the speed of the proposed image encryption technique on an Intel Core I3 CPU 2.3 GHz and 3.99 GB of RAM running on Windows XP and MATLAB 7.1 programming. For accuracy each set of the timing tests was executed several times for considerable number of images and then the average obtained was reported. In Table 5, we can see the comparison results for the proposed scheme and other schemes. Table 5 shows that the proposed algorithm is fast compared to the other schemes.

4.5 Information Entropy

As one of the most important features, the information entropy is often used to measure the randomness of the cipher-image. The entropy H(s) of a message source is given by:

Table 4 Comparison of the average $NPCR_{R,\ G,\ B}$ and $UACI_{R,\ G,\ B}$ values

Algorithm	Average ($NPCR_{R,\ G,\ B}$)	Average ($UACI_{R,\ G,\ B}$)
Proposed	0.996831	0.334412
[34]	0.996168	0.334659
[35]	0.989182	0.327865
[26]	0.996355	0.334188
[36]	0.9965	0.3348
[37]	0.9982	0.3346

Table 5 Comparison of encryption speeds for the proposed scheme and different schemes

Algorithm	Speed (M bit/s)
Proposed	9.16
[26]	8.11
[40]	8.16
[41]	9.41
[42]	5.15

$$H(s) = -\sum_{i=0}^{2^n-1} p(s_i)\log_2 p(s_i), \tag{28}$$

where $p(s_i)$ represents the probability of the symbol s. The entropy should ideally be H(s) = 8 for a cipher-image with 2^8–1 gray levels, which shows that the information is random. In the paper the information entropy of the cipher-image is 7.9973, close to the ideal value 8. Hence, we conclude that the proposed algorithm has high randomness.

5 Conclusions

This paper has realized the quantum image encryption and decryption. Image information is ciphered by the proposed encryption algorithm based on general

Arnold transform with keys and quantum chaotic map. By improving the Arnold transform algorithm, we not only enlarge the key space to resist against any key sensitivity and any brute-force attack, but also raise the running speed of the process of the encryption. The experiment shows that only one time general Arnold transform with keys has a good result. In order to enhance the sensitivity of the cryptosystem, the key generator apply the addressing map to get initial conditions and parameters. Quantum chaotic sequence possesses perfect chaotic character, which is used to change the pixel values of the plain-image by "folding the picture". The experimental results demonstrate that the folding algorithm can achieve sensitivity to initial values, robustness, resistance against common attacks, large key space and possesses the high encryption speed (speed $>$ 9.16 M bit/s). Accordingly the proposed algorithm is suitable to practical uses to protect the digital image information over the Internet.

Acknowledgements This work was financially supported by self-determined research funds of CCNU from the colleges' basic research and operation of MOE (Grant No. CCNU15GF007).

References

1. Liu, H., & Wang, X. (2010). Color image encryption based on one-time keys and robust chaotic maps. *Computers & Mathematics with Applications, 59*(10), 3320–3327.
2. Diaconu, A. V., Ionescu, V., & Iana, G. (2016). A new bit-level permutation image encryption algorithm. In International Conference on Communications (pp. 411–416).
3. Liu, H., & Wang, X. (2011). Color image encryption using spatial bit-level permutation and high-dimension chaotic system. *Optics Communications, 284*(16–17), 3895–3903.
4. Kumar, M., & Vaish, A. (2016). An efficient encryption-then-compression technique for encrypted images using SVD. *Digital Signal Processing, 60,* 81–89.
5. Zhang, Y. (2014). Cryptanalysis of an image encryption algorithm based on chaotic modulation of Arnold dual scrambling and DNA computing. *Advanced Science Focus, 2*(1), 67–82.
6. Wan, R., Mo, H., & Yu, S. (2014). Document and image encryption based on OTP optimized by hyper-chaos mapping DNA computing. *Computer Measurement & Control, 22*(10), 3278–3281.
7. Zhou, S., Wang, B., Zheng, X., & Zhou, C. (2016). An image encryption scheme based on DNA computing and cellular automata. *Discrete Dynamics in Nature and Society, 2016*(2), 1–9.
8. Abbas, A. M. (2015). Image encryption based on independent component analysis and Arnold's cat map. *Egyptian Informatics Journal, 17*(1), 139–146.
9. Guo, Q., Liu, Z., & Liu, S. (2010). Color image encryption by using Arnold and discrete fractional random transforms in IHS space. *Optics and Lasers in Engineering, 48*(12), 1174–1181.
10. Zhou, N. R., Hua, T. X., Gong, L. H., Pei, D. J., & Liao, Q. H. (2015). Quantum image encryption based on generalized Arnold transform and double random-phase encoding. *Quantum Information Processing, 14*(4), 1193–1213.
11. Sui, L., & Gao, B. (2013). Color image encryption based on gyrator transform and Arnold transform. *Optics & Laser Technology, 48*(6), 530–538.
12. Chen, L., Zhao, D., & Ge, F. (2013). Image encryption based on singular value decomposition and Arnold transform in fractional domain. *Optics Communications, 291*(291), 98–103.
13. Das, P., Kushwaha, S. C., & Chakraborty, M. (2015). Multiple embedding secret key image steganography using LSB substitution and Arnold transform. In International Conference on Electronics and Communication Systems (pp. 845–849).
14. Jin, C., & Tu, Z. W. (2016). A novel color image encryption algorithm using chaotic map and improved RC4. *Advances in Intelligent Systems and Computing, 466,* 3–14.
15. Wang, X. Y., & Wang, M.-J. (2010). Projective synchronization of nonlinear-coupled spatiotemporal chaotic systems. *Nonlinear Dynamics, 62*(3), 567–571.
16. Wang, X.-Y., Yang, L., & Liu, R. (2010). A chaotic image encryption algorithm based on perceptron model. *Nonlinear Dynamics, 62*(3), 615–621.
17. Liu, H., Wang, X., & Kadir, A. (2012). Image encryption using DNA complementary rule and chaotic maps. *Applied Soft Computing, 12*(5), 1457–1466.
18. Baptista, M.-S. (1998). Cryptography with chaos. *Physics Letters A, 240*(1–2), 50–54.
19. Fridrich, J. (2011). Symmetric ciphers based on two-dimensional chaotic maps. *International Journal of Bifurcation & Chaos, 8*(6), 1259–1284.
20. Wang, X., & Guo, K. (2014). A new image alternate encryption algorithm based on chaotic map. *Nonlinear Dynamics, 76*(4), 1943–1950.
21. Tang, Z., Zhang, X., & Lan, W. (2015). Efficient image encryption with block shuffling and chaotic map. *Multimedia Tools and Applications, 74*(15), 5429–5448.
22. Jawad, L. M., & Sulong, G. (2015). Chaotic map-embedded blowfish algorithm for security enhancement of colour image encryption. *Nonlinear Dynamics, 81*(4), 2079–2093.
23. Akhshani, A., Akhavan, A., Mobaraki, A., Lim, S. C., & Hassan, Z. (2014). Pseudo random number generator based on quantum chaotic map. *Communications in Nonlinear Science and Numerical Simulation, 19*(1), 101–111.
24. Akhshani, A., Akhavan, A., Lim, S. C., & Hassan, Z. (2012). An image encryption scheme based on quantum logistic map. *Communications in Nonlinear Science and Numerical Simulation, 17*(12), 4653–4661.
25. Seyedzadeh, S. M., Norouzi, B., Mosavi, M. R., & Mirzakuchaki, S. (2015). A novel color image encryption algorithm based on spatial permutation and quantum chaotic map. *Nonlinear Dynamics, 81*(1–2), 1–19.
26. Abd El-Latif, A. A., Li, L., Wang, N., Han, Q., & Niu, X. (2013). A new approach to chaotic image encryption based on quantum chaotic system, exploiting color spaces. *Signal Processing, 93*(11), 2986–3000.

27. Wang, X., & Shi, Q. J. (2005). New type crisis, hysteresis and fractal in coupled logistic map. *Chinese Journal of Applied Mechanics, 4,* 501–506.
28. Wang, X. Y., Zhang, Y. Q., & Zhao, Y. Y. (2015). A novel image encryption scheme based on 2-d logistic map and DNA sequence operations. *Nonlinear Dynamics, 82*(3), 1269–1280.
29. Sun, X. H. (2013). *Image encryption algorithms and practices with implementations in C#*. Beijing: Science Press.
30. Khan, M., Shah, T., & Batool, S. I. (2014). Texture analysis of chaotic coupled map lattices based image encryption algorithm. *3D Research, 5*(3), 1–5.
31. Zhang, Y. Q., & Wang, X. Y. (2014). Spatiotemporal chaos in mixed linear-nonlinear coupled logistic map lattice. *Physica A, 402*(10), 104–118.
32. Wang, X., Teng, L., & Qin, X. (2012). A novel colour image encryption algorithm based on chaos. *Signal Processing, 92*(4), 1101–1108.
33. Zhang, Y. Q., & Wang, X. Y. (2014). A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice. *Information Sciences, 273*(8), 329–351.
34. Hanchinamani, G., & Kulkarni, L. (2015). An efficient image encryption scheme based on a peter De Jong chaotic map and aRC4 stream cipher. *3D Research, 6*(3), 1–15.
35. Rehman, A. U., Khan, J. S., Ahmad, J., & Hwang, S. O. (2016). A new image encryption scheme based on dynamic s-boxes and chaotic maps. *3D Research, 7*(1), 1–8.
36. Wang, X. Y., Zhang, Y. Q., & Bao, X. M. (2015). A novel chaotic image encryption scheme using DNA sequence operations. *Optics and Lasers in Engineering, 73*(3), 53–61.
37. Wang, X. Y., Gu, S. X., & Zhang, Y. Q. (2015). Novel image encryption algorithm based on cycle shift and chaotic system. *Optics and Lasers in Engineering, 68,* 126–134.
38. Jorgensen, P. (2015). Applied cryptography: Protocols, algorithm, and source code in c. *Government Information Quarterly, 13*(3), 336.
39. Norouzi, B., Seyedzadeh, S. M., Mirzakuchaki, S., & Mosavi, M. R. (2015). A novel image encryption based on row-column, masking and main diffusion processes with hyper chaos. *Multimedia Tools and Applications, 74*(3), 781–811.
40. Patidar, V., Pareek, N. K., Purohit, G., & Sud, K. K. (2010). Modified substitution-diffusion image cipher using chaotic standard and logistic maps. *Communications in Nonlinear Science and Numerical Simulation, 15*(15), 2755–2765.
41. Hua, Z., Zhou, Y., Pun, C. M., & Chen, C. L. P. (2015). 2D sine logistic modulation map for image encryption. *Information Sciences, 297*(11), 80–94.
42. Mazloom, S., & Eftekhari-Moghadam, A. M. (2009). Color image encryption based on coupled nonlinear chaotic map. *Chaos, Solitons & Fractals, 42*(3), 1745–1754.