IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 3, 2019, accepted July 18, 2019, date of publication July 29, 2019, date of current version August 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2931732

An Image Segmentation Encryption Algorithm Based on Hybrid Chaotic System

ZHENLONG MAN[1,2], JINQING LI[1,2], XIAOQIANG DI[1,2,3], AND OU BAI[4], (Member, IEEE)

[1]School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
[2]Jilin Province Key Laboratory of Network and Information Security, Changchun 130022, China
[3]Information Center, Changchun University of Science and Technology, Changchun 130022, China
[4]Department of Electrical and Computer Engineering, Florida International University, Miami, FL 33199, USA

Corresponding authors: Jinqing Li (lijinqing@cust.edu.cn) and Xiaoqiang Di (dixiaoqiang@cust.edu.cn)

This work was supported in part by the China's Natural Science Foundation Project (20190201188JC) of the Science and Technology Department of Jilin Province and in part by the China's Science and Technology Project (JJKH20181137KJ) for the 13th Five-Year Plan of the Education Department of Jilin Province.

ABSTRACT Image encryption is an effective technology to protect digital image confidentiality. This paper presents an image segmentation encryption algorithm based on a hybrid chaotic system. First, a chaotic sequence is obtained by iterating a Quantum Cellular Neural Network (QCNN), and then it is scrambled by a 4-D hyperchaotic system to generate a key pool. Second, the chaotic pointers generated by 3-D chaotic systems and QCNN with different initial values are used to get the keys for image segmentation, scrambling, and diffusion from the key pool. Then, the plain-image is divided into two blocks by the chaotic segmentation method and scrambled by intra-block and inter-block pixel exchange. In addition, two blocks are statically diffused, and the cipher-image is obtained by dynamic diffusing after combining the image blocks. Especially, the key pool increases the efficiency of the proposed algorithm, and chaotic segmentation reduces the cipher-image pixel correlation. Finally, the simulation results and performance analysis indicate that the proposed algorithm has a well-security, high sensitivity, and faster speed.

INDEX TERMS Chaotic pointer, chaotic segmentation, hybrid chaotic system, quantum cellular neural network.

I. INTRODUCTION

With the rapid development of the internet, the number of digital images are transmitted over public and shared networks keeps increasing. The security of digital images has become a serious issue which has gained a lot of attention because the private information contained in an image can be intercepted, tampered and destroyed illegally. Image encryption is an effective way to protect the image transmission. Unlike text encryptions, there are some unique characteristics, such as bulk capacities, strong correlations between the adjacent pixels, and high redundancy [1]. Hence, the image encryption is a challenging task.

A lot of encryption schemes have been proposed based on different technologies, such as breadth-first search [2], edge password [3], elliptic curve [4], fractional Fourier transform [5] and chaos [6]–[12]. Among these technologies, chaotic-based schemes are one of the most effective encryption methods due to their complexity and non-linearity, high sensitivity of initial conditions and control parameters, non-periodicity and pseudo-randomness [13]. Therefore, since Matthews first proposed the chaotic encryption algorithm in 1989 [14], many image encryption schemes based on low-dimensional chaos have been developed [15]–[18]. Most existing low-dimensional chaotic schemes have two issues. One is that their dynamic properties may degrade due to the finite precision of computer implementation [19]. Another is low dimensional schemes lead to a smaller key space [19]. Therefore, more attentions have been paid to high-dimensional chaotic schemes for its larger key space, complex and unpredictable nonlinear behavior [20]–[24], especially the well-security hyper-chaotic image encryption [25]. Most of the existing image encryption schemes are composed of the diffusion stage and the permutation stage [26]. In the permutation stage, the position of image pixels is changed, which makes visual confusion. In the diffusion stage, it is required to extend the influence of a single plain pixel or key to as more cipher pixels as possible

The associate editor coordinating the review of this manuscript and approving it for publication was Constantinos Marios Angelopoulos.

to mask the relationships of statistical properties between cipher-images and plain-images. Usually, diffusion and permutation are performed for the whole image. In order to improve permutation effect, the plain-image is firstly divided into several equal blocks in vertical, horizontal, or diagonal directions [27]–[29], and then pixels exchange between intra-block and inter-block. The rules for the exchange of pixels are controlled by a hyper chaotic system [27], the cat mapping [28] and the logistic mapping [29], respectively. The QCNN is one of the hyper-chaotic systems, and it has been constructed with the Quantum Cell Automata (QCA) based on the Schrodinger equation and the Chua's cellular neural network [30]. It can obtain complex linear dynamic characteristics from the polarizability and quantum phase of each QCA [31]. Compared with other chaotic systems, QCNN supports ultra-high integration density, ultra-low power consumption, real-time signal processing and parallel computing [32]. The chaotic properties of QCNN are analyzed in the literature [33]–[35] for more details, especially it can avoid the periodic window problem. Therefore, QCNN is more suitable for image encryption than low-dimensional chaotic schemes [36].

In order to furtherly enhance our algorithm security, a QCNN sequence is used for the plain-image segmentation, and it is different from segmentation [34]–[36] in horizontal/vertical/diagonal directions. Compared with [34]–[36], our chaotic segmentation can reduce the correlation among pixels even more. Firstly, a chaotic sequence is generated by a QCNN for iterations several times, and it is regarded as the key pool, so that it avoids multiple iterations to improves the efficiency of key generation. Then a 4-D hyper chaotic index is used to scramble the chaotic sequence to strengthen the key‘s randomness. The chaotic pointer generated by another QCNN and a 3-D chaotic system is regarded as indexes to get keys for image segmentation, pixel exchange rule, scrambling, and diffusion from the key pool. After image segmentation, blocks are scrambled by pixel exchange rules within the intra-block and inter-block. They are diffused by combining static and dynamic methods. The simulation results and security analysis, such as violent attack detection, differential attack and shear attack, verify the feasibility of the scheme. The performance is tested in terms of pixel correlation, information entropy and encryption speed. From the analysis results, we can see that our scheme has higher encryption efficiency and better security.

The rest of this paper is structured as follows. Section 2 briefly introduces the chaotic systems of the proposed algorithm, the chaotic pointer generation, and the chaotic segmentation method. In Section 3, the encryption and decryption algorithm are described in detail. Section 4 analyzes the performance of the proposed algorithm through various statistical and security tests. The conclusion is given in Section 5.

II. PREPARATORY WORK

In this section, we introduce the preparatory work, such as the generation of the quantum chaotic key pool, the function of the chaotic pointer, the method of chaotic image segmentation and so on.

A. TWO CELL QUANTUM NEURAL NETWORK HYPERCHAOTIC SYSTEM

Lent [37] proposes the QCA, which can construct a locally coupled QCNN [38]. For a quantum neural network coupled with two cells, the state equation of the chaotic system is presented by

$$\begin{cases}\dot{p}_1 = -2\omega_{01}\sqrt{1-p_1^2}\sin\varphi_1 \\ \dot{\varphi}_1 = -\omega_{02}p_1 - p_2 + 2\omega_{01}p_1\cos\varphi_1/(\sqrt{1-p_1^2}) \\ \dot{p}_2 = -2\omega_{03}\sqrt{1-p_2^2}\sin\varphi_2 \\ \dot{\varphi}_2 = -\omega_{04}(p_2-p_1) + 2\omega_{03}p_2\cos\varphi_2/(\sqrt{1-p_2^2})\end{cases} \tag{1}$$

where: p_1, p_2 is the polarizability; φ_1, φ_2 is the quantum phase; ω_{01}, ω_{03} is the proportional coefficient of the energy between the points in each cell; ω_{02}, ω_{04} is the weighted influence factor for the difference between the polarizabilities of adjacent cells. When $\omega_{01} = \omega_{03} = 0.28$, $\omega_{02} = 0.7$, and $\omega_{04} = 0.3$, the system is in a chaotic state. Part of its attractors is shown in (a) (b) (c) (d) of Fig.1.

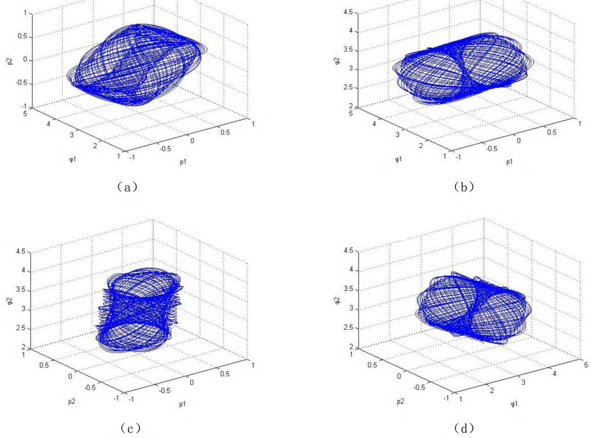


FIGURE 1. Attractor of hyperchaotic system of QCNN.

Dynamic behavior of the QCNN system can be analyzed by calculating the lyapunov exponent λ. When $\omega_{01} = \omega_{03} = 0.28$, $\omega_{02} = 0.7$, $\omega_{04} \in [0, 1]$, the lyapunov exponent is shown in Fig.2. It can be observed that when $\omega_{04} > 0.1$, the QCNN has three positive lyapunov exponents, indicating that it is hyperchaotic system.

B. FOUR-DIMENSIONAL HYPERCHAOTIC SYSTEM

The 4-D hyperchaotic system formula is as follows [39]:

$$\begin{cases}\dot{x}_1 = \delta_1(x_2 - x_1) \\ \dot{x}_2 = \delta_2 x_1 + \delta_3 x_2 - x_1 x_3 + x_4 \\ \dot{x}_3 = x_2^2 - \delta_4 x_3 \\ \dot{x}_4 = -\delta_5 x_1\end{cases} \tag{2}$$

Let $x = [x_1, x_2, x_3, x_4]$ represent the state vector of the system. δ_1, δ_2, δ_3, δ_4, δ_5 are system parameters.

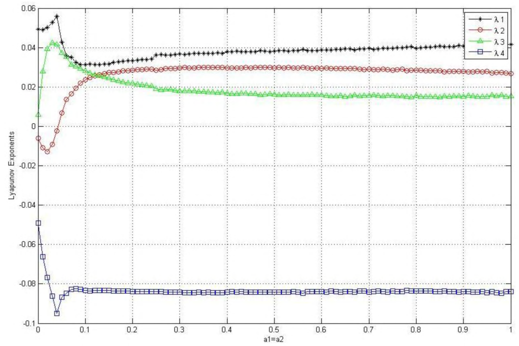
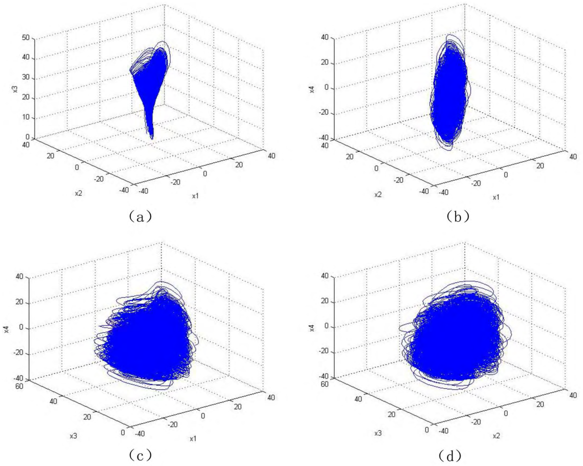


FIGURE 2. The Lyapunov exponents of QCNN.



(a) (b) (c) (d)

FIGURE 3. Attractor of 4D hyperchaotic system.

When $\delta_1 = 27.5$, $\delta_2 = 3$, $\delta_3 = 19.3$, $\delta_4 = 2.9$, $\delta_5 = 3$, the system is hyperchaotic. Lyapunov exponent $\lambda 1 = 1.6170$, $\lambda 2 = 0.1123$, $\lambda 3 = 0$, $\lambda 4 = -12.8245$. The system has two positive Lyapunov exponents, indicating that it is a hyperchaotic system at this time [40]. Some of its attractors are shown in (a) (b) (c) (d) of Fig.3.

C. THREE-DIMENSIONAL CHAOTIC SYSTEM

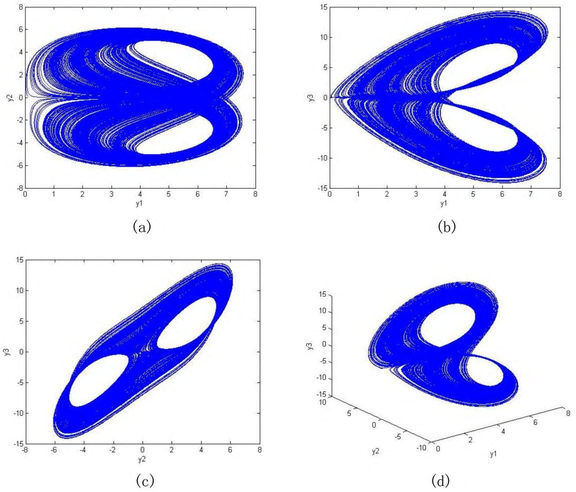
The 3-D chaotic system formula is as follows [41]:

$$\begin{cases} \dot{y}_1 = -ay_1 + y_2y_3 \\ \dot{y}_2 = by_2 - y_1y_3 - y_3 \\ \dot{y}_3 = -cy_3 + y_2^3 \end{cases} \tag{3}$$

Parameters a, b, and c are real constants, when $a = 3$, $b = 5$, and $c = 10$, the system is chaotic. Lyapunov exponent $\lambda 1 = 0.03$, $\lambda 2 = -0.01$ and $\lambda 3 = -7.78$. There are positive exponents in Lyapunov exponent, so the system has chaotic characteristics. Its attractors are shown in (a) (b) (c) (d) of Fig.4.

D. QUANTUM CHAOTIC KEY POOL AND CHAOTIC POINTER GENERATION SCHEME

The 4-D hyperchaotic system is used to scramble the sequence generated by the QCNN to obtain the quantum chaotic key pool. Then the 3-D chaotic system and another QCNN are combined together to generate chaotic pointers. The encryption key is selected from the key pool by the chaotic pointer.



(a) (b) (c) (d)

FIGURE 4. Attractor of 3-D chaotic system.

1) QUANTUM CHAOTIC KEY POOL

System (1) is iterated $t1$ times with the initial values $p_1(0)$, $p_2(0)$, $\varphi_1(0)$, $\varphi_2(0)$. We can get four quantum chaotic sequences $Qcn1$, $Qcn2$, $Qcn3$, $Qcn4$. Four hyperchaotic sequences are joined from front to back to form a 1-D sequence QC with the length of $4t1$. The quantum chaotic key pools $QKP0$, $QKP1$ and $QKP2$ are obtained by

$$\begin{cases} QKP0 = floor((QC) \times 10^{14}) \bmod G + 1 \\ QKP1 = floor(abs(QC) \times 10^{14}) \bmod (G-1) + 1 \\ QKP2 = QKP1 \bmod G/2 + 1 \end{cases} \tag{4}$$

When the image is an 8-bit grayscale image, $G = 256$.

We take $x_1(0), x_2(0), x_3(0), x_4(0)$ as initial values, iterate the system (2) $t1$ times, and get four hyperchaotic sequences $H1$, $H2$, $H2$, $H4$. The four hyperchaotic sequences are spliced from front to back into a 1-D sequence HC of length $4t1$.The hyperchaotic sequence $HC1$ is used as the index to scramble the quantum chaotic sequence $QKP1$.

$$HC1 = floor(HC \times 10^{14}) \bmod (4t1) + 1 \tag{5}$$

$$\begin{cases} \gamma = QKP1(HC1(i)) \\ QKP1(HC1(i)) = QKP1(HC1(i+1)) \\ QKP1(HC1(i+1)) = \gamma \end{cases} \tag{6}$$

where γ is the intermediate variable, $i = 1, 2, 3, \ldots, 4 \times t1 - 1$. The scrambling process is illustrated in Fig. 5

2) CHAOTIC POINTER

Initial values $y_1(0), y_2(0), y_3(0)$, system (3) is iterated $t1$ times to produce the chaotic sequences $C1$, $C2$, and $C3$. The initial values are $p_{11}(0)$, $p_{21}(0)$, $\varphi_{11}(0)$ and $\varphi_{21}(0)$, and the four quantum chaotic sequences $NQ1$, $NQ2$, $NQ3$ and $NQ4$ are

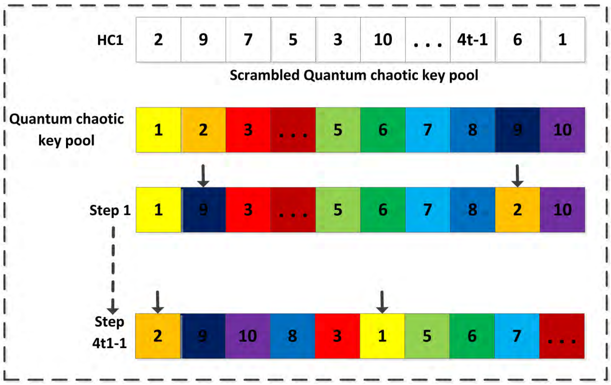


FIGURE 5. Scrambling of Quantum Chaotic Key Pool.

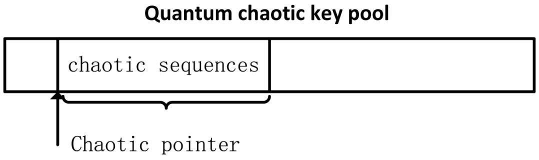


FIGURE 6. Chaotic pointer scheme.

obtained by iterating system (1) $t1$ times. The new chaotic sequences $CN1$, $CN2$ and $CN3$ can be obtained by

$$\begin{cases} CN1 = C1 \bmod NQ1 \\ CN2 = C2 \bmod NQ2 \\ CN3 = C3 \bmod NQ3 \end{cases} \tag{7}$$

The chaotic sequence $Sp1 - Sp6$ is obtained by $QC1$, $QC2$ and $QC3$ respectively.

$$\begin{cases} QC1 = floor(CN1 \times 10^{14}) \bmod length(QKP1) + 1 \\ Sp1 = QC1(length(QC1)) \\ Sp2 = QC1(length(QC1) - M) \\ QC2 = floor(CN2 \times 10^{14}) \bmod length(QKP1) + 1 \\ Sp3 = QC2(length(QC2)) \\ Sp4 = QC2(length(QC2) - M) \\ QC3 = floor(CN3 \times 10^{14}) \bmod length(QKP1) + 1 \\ Sp5 = QC3(length(QC3)) \\ Sp6 = QC3(length(QC3) - M) \end{cases} \tag{8}$$

where M is the length or width of the image.

Generate chaotic pointers $QSp1$, $QSp2$, $QSp3$, $QSp4$, $QSp5$, $QSp6$, as shown in (9). The role of chaotic pointers is shown in Fig.6.

$$\begin{cases} QSp1 = (bitxor(Sp1, Sp3) + Sp5) \bmod length(QKP1) \\ QSp2 = (bitxor(Sp1, Sp4) + Sp6) \bmod length(QKP1) \\ QSp3 = (bitxor(Sp3, Sp5) + Sp1) \bmod length(QKP1) \\ QSp4 = (bitxor(Sp3, Sp5) + Sp2) \bmod length(QKP1) \\ QSp5 = (bitxor(Sp5, Sp1) + Sp3) \bmod length(QKP1) \\ QSp6 = (bitxor(Sp5, Sp2) + Sp4) \bmod length(QKP1) \end{cases} \tag{9}$$

E. QUANTUM CHAOTIC IMAGE SEGMENTATION

The preprocessing operation is performed before the plain-image is encrypted. It is assumed that the size of the plain-image is $M \times N$, and M is an even number. A new quantum chaotic sequence S is selected by the chaotic pointer $QSp1$ in the key pool $QKP0$.

$$\begin{cases} Q0 = [QKP0(QSp1 : end), QKP0(1 : QSp1 - 1)] \\ S = Q0(1 : nr) \end{cases} \tag{10}$$

where nr is a number less than the length of the quantum chaotic key pool and ensures that there are no duplicates in the mapped Cs. Chaotic Segmentation Sequence Css of image is given by

$$\begin{cases} Cs = ceil(S \times 10^5) \bmod M \\ Css(u) = \sim ismember(Css(u), Cs(l)) \end{cases} \tag{11}$$

where $u = 1, 2, 3, \ldots, M, l = 1, 2, 3, \ldots, M/2 \times N/2$. $\sim$ $ismember$: represents a function to remove duplicate elements.

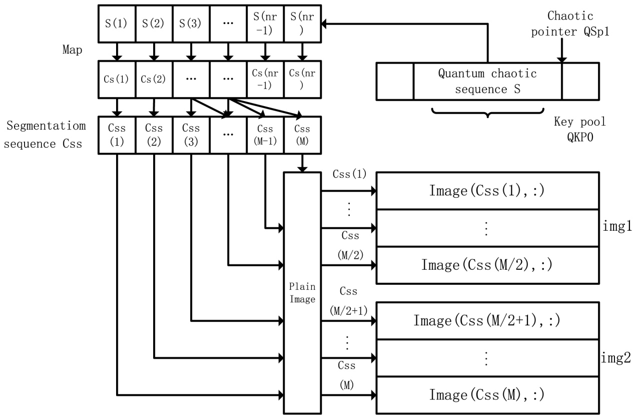


FIGURE 7. Quantum chaotic image segmentation method.

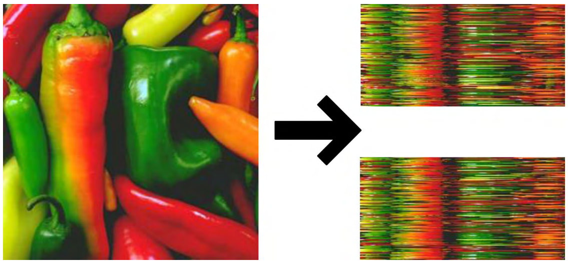


FIGURE 8. Chaotic segmentation image.

The plain-image is split to obtain image block $img1$ and image $img2$ using quantum chaotic image segmentation method. The segmentation process is illustrated in Fig 7, and the split results of "peppers" are shown in Fig 8. The elements in the chaotic segmentation sequence Css are used to represent the line numbers in the plain-image. The odd lines of all the Css in the plain-image are extracted to form

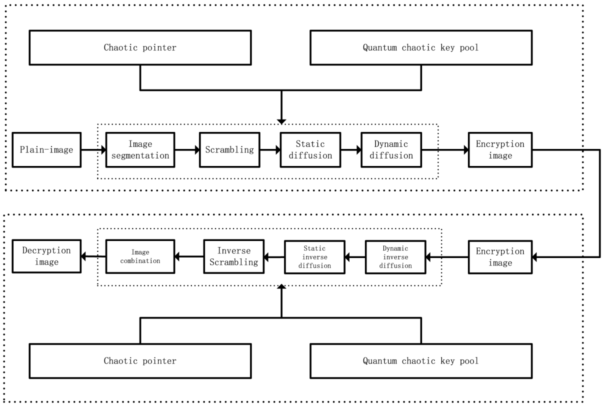


FIGURE 9. Image encryption and decryption block diagram.

the image block *img*1, and then the even lines constitute the image block *img*2.

III. IMAGE ENCRYPTION/DECRYPTION SCHEME

A. IMAGE ENCRYPTION SCHEME

An image segmentation encryption algorithm based on the hybrid chaotic system is proposed. This algorithm can be used to encrypt the color image and gray image. For a color image, the value of its color component is processed separately, and for a gray image, the gray value is processed. Its block diagram is given in Fig.9, and the encryption flow chart is presented in Fig.10.

Step 1: The size of the plain-image is $M \times N$. According to the method in Section II.E. the plain-image is split to *img*1 and *img*2.

Step 2: Quantum Control Table QCT is used to control the scrambling mode of the pixels in the quantum exchange table in step 3. The QCT with $M/2 \times N$ size is constructed by chaotic matrix $CT1$ and $CT2$.

$$QCT = bitxor(CT1, CT2) \bmod ku_1 \tag{12}$$

where $ku_1 \in [1, M \times N]$, $ku_1 \in \mathbb{N}$, ku_1 is the control parameter of the QCT. $CT1$ and $CT2$ are obtained from the follow:

$$\begin{cases} Q1 = [QKP2(QSp1 : end), QKP2(1 : QSp1 - 1)] \\ CT1 = reshape(Q1(1 : M/2 \times N), M/2, N) \\ Q2 = [QKP2(QSp2 : end), QKP2(1 : QSp2 - 1)] \\ CT2 = reshape(Q2(1 : M/2 \times N), M/2, N) \end{cases} \tag{13}$$

Step 3: Quantum Exchanging Tables $QEXT$ and $QEYT$ are used to exchange pixels in two image blocks. The $QEXT$ and $QEYT$ are constructed by chaotic matrix XT and YT.

$$\begin{aligned} &QEXT(i,j) \\ &\quad = \begin{cases} (XT(i,j) + floor(abs(ku_2 \times 10^7)) \bmod (M/2 \times N) \\ \bmod M/2), \qquad abs(XT(i,j) - i) < M/8 \\ XT(i,j), \qquad others \end{cases} \\ &QEYT(i,j) \\ &\quad = \begin{cases} YT(i,j) \bmod N, & abs(YT(i,j) - i) < N/4 \\ YT(i,j), & others \end{cases} \end{aligned} \tag{14}$$

where $i = 1, 2, 3, \ldots, M/2, j = 1, 2, 3, \ldots, N$. ku_2 is the control parameter of the $QEXT$. XT and YT are obtained from the following formula:

$$\begin{cases} Q3 = [QKP2(QSp3 : end), QKP2(1 : QSp3 - 1)] \\ XT = reshape(Q3(1 : M/2 \times N), M/2, N) \\ Q4 = [QKP2(QSp4 : end), QKP2(1 : QSp4 - 1)] \\ YT = reshape(Q4(1 : M/2 \times N), M/2, N) \end{cases} \tag{15}$$

Step 4: Scrambling image blocks *img*1 and *img*2 according to the "inter-intra-block" method described below, as shown in Fig 11. According to the value of QCT, there are two modes for scrambling the pixels in the two image blocks *img*1 and *img*2, which are the "inter-block exchange mode" and the "intra-block exchange mode", respectively.

When the quantum control table $QCT(i,j) = 0$, $img1(i,j)$ exchange with $img1(QEXT(i,j), QEYT(i,j))$; When $QCT(i,j) > 0$, $img1(i,j)$ and $img2(QEXT(i,j), QEYT(i,j))$ are exchanged. When the quantum control table $QCT(i,j) = 0$, $img2(i,j)$ exchange with $img2(QEXT(i,j), QEYT(i,j))$; When $QCT(i,j) > 0$, $img2(i,j)$ and

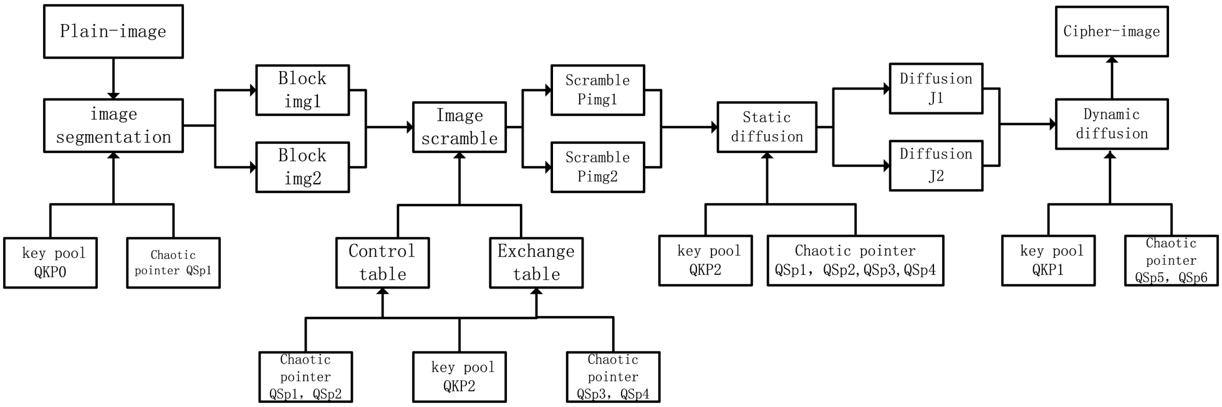


FIGURE 10. Image encryption flow chart.

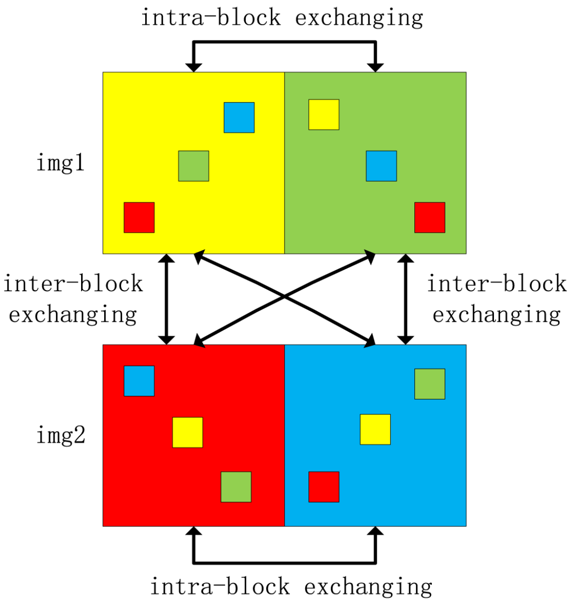


FIGURE 11. "inter-intra-block" scrambling.

$img1(QEXT(i,j), QEYT(i,j))$ are exchanged. The scrambled image blocks $Pimg1$ and $Pimg2$ are obtained after traversing all elements of QCT. The scrambling process is shown in Fig.11.

Step 5: The static diffusion of the scrambled image blocks $Pimg1$ and $Pimg2$ is as follows:

$$\begin{cases} J1(i,j) = bitxor(CT1(i,j), bitxor(CT(i,j), Pimg1(i,j))) \\ J2(i,j) = bitxor(YT(i,j), bitxor(XT(i,j), Pimg2(i,j))) \end{cases} \tag{16}$$

where $i = 1, 2, 3, 4, \ldots, M/2 \times N. j = M/2 \times N, M/2 \times N - 1, \ldots, 3, 2, 1$. $J1$ and $J2$ are the static diffusion results of $Pimg1$ and $Pimg2$ respectively.

Step 6: $J1$ and $J2$ are spliced to obtain a static diffusion sequence CJ of length $M \times N$:

$$\begin{aligned} CJ = [&J1(1), J1(2), \ldots, J1(M/2 \times N), \\ &J2(1), J2(2), \ldots, J2(M/2 \times N)] \end{aligned} \tag{17}$$

Step 7: The dynamic chaotic key streams $QK1$ and $QK2$ are generated using the quantum chaotic key pool $QKP1$ and the chaotic pointers $QSp5$ and $QSp6$, as shown by

$$\begin{cases} Q5 = [QKP1(QSp5 : end), QKP1(1 : QSp5 - 1)] \\ QK1 = Q5(1, M \times N) \\ Q6 = [QKP1(QSp6 : end), QKP1(1 : QSp6 - 1)] \\ QK2 = Q6(1, M \times N) \end{cases} \tag{18}$$

Step 8: CJ is dynamically diffused by $Qk1$ and $QK2$ using the manner described by Eq (19), and the dynamic diffusion sequence D_{CJ} is obtained.

$$\begin{cases} R = bitxor(ku_3, QK1(M \times N)) \\ R1 = bitxor(CJ(k), QK1(k)) \\ R2 = bitxor(mod(R + QK1(k), M), QK2(k)) \\ D_{CJ}(k) = bitxor(R1, R2) \end{cases} \tag{19}$$

where $k = 1, 2, 3, \ldots, M \times N$, $ku_3 \in \mathbb{N}$, ku_3 is the control parameter for the dynamic diffusion.

Step 9: The dynamic encryption sequence D_{CJ} is converted into a matrix from top to bottom and left to right to obtain the encrypted image $Cimage$.

B. DECRYPTION PROCESS

Since the encryption algorithm is a symmetric encryption algorithm, the decryption algorithm is the inverse process of the encryption algorithm.

IV. SECURITY AND PERFORMANCE ANALYSIS

In this section, we discuss the performance of the proposed algorithm. We choose the standard color plain-image "sailboat" and "pepper" with the size of 256 × 256 × 3 as the testing subject. The initial keys are:$p_1(0) = 0.15$, $p_2(0) = 4.89$, $1(0) = 0.21$, $2(0) = 3.12$, $x_1(0) = 2.55$, $x_2(0) = 5.2$, $x_3(0) = 3.12$, $x_3(0) = 7.31$, $p_12(0) = 0.189$, $p_22(0) = 4.67$, $12(0) = 0.198$, $22(0) = 3.22$, $y_1(0) = -1$, $y_2(0) = 0$, $y_3(0) = 1$, $ku_1 = 7$, $ku_2 = 1$, $ku_3 = 10$, Fig.12(a), 12(d) denote plain-images during the experiments. Fig.12(b), 12(e) are cipher-images which are completely invisible. Fig.12(c), 12(f) are decrypted images which are identical to plain-images.

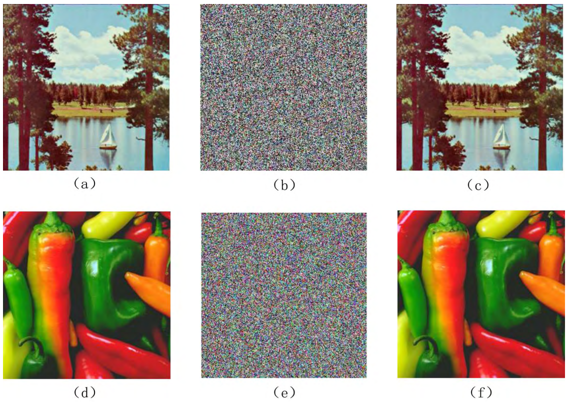


FIGURE 12. Experimental results: (a)(d) is the plain-image of "sailboat" and "pepper", (b) (e) is the cipher-image of "sailboat" and 'pepper", (c) (f) is the decryption image of "sailboat" and "pepper".

A. KEY SPACE

The ideal image encryption algorithm should have a key space greater than 2^{100} [42], providing high degree of security against resistant to brute force attacks. The keys:p_1, p_2, φ_1, φ_2, ω_{01}, ω_{02}, ω_{03}, ω_{04}, x_1, x_2, x_3, x_4, δ_1, δ_2, δ_3, δ_4, δ_5, p_{12}, p_{22}, φ_{12}, φ_{22}, ω_1, ω_2, ω_3, ω_4, y_1, y_2, y_3, a, b, c. Because each precision of the initial key and parameter is 10^{-16}. The key

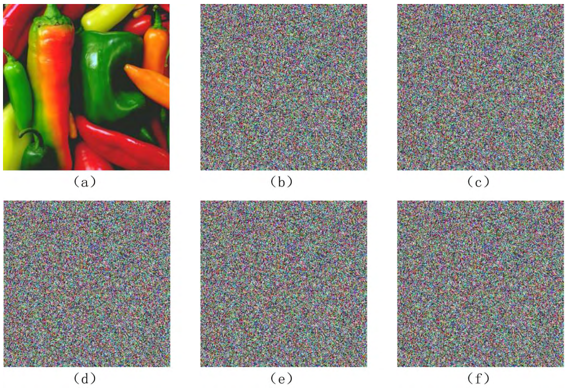


FIGURE 13. Key sensitivity test: (a) "pepper" plain-image, (b) cipher-image using the original key, (c) incorrect decryption using $p_1 + 10^{-16}$, (d) incorrect decryption using $x_1 + 10^{-16}$, (e) incorrect decryption using $p_{12} + 10^{-16}$, (f) incorrect decryption using $y_1 + 10^{-16}$.

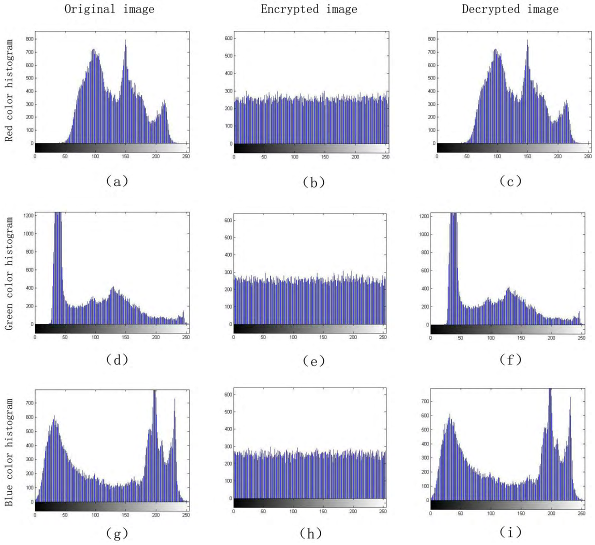


FIGURE 14. Histogram analysis: (a) (d) (f) are histograms of the three color components of the plain-image of "sailboat", (b) (d) (j) are histograms of the three color components of the cipher-image of "sailboat", (c) (e) (i) are histograms of the three color components of the decrypted image of "sailboat".

space is $(10^{16})^{31} = 10^{496} > 2^{1488}$. Therefore, the key space is large enough to resist all types of brute force attacks [43].

B. KEY SENSITIVITY ANALYSIS

A secure cryptosystem should be sensitive to the key, a slight change in the encryption key will result in a very different encrypted image, a slight change in the decryption key will not decrypt the image. Taking a 256 × 256 "pepper" image as an example, if the small change of 10^{-16} is shifted in keys p_1, x_1, p_{12} and y_1 respectively, the decryption result will be incorrect, as shown in Fig.13(a)-(f). This shows the high sensitivity of the key to our algorithm.

C. HISTOGRAM ANALYSIS

As we all know, the image histogram represents the distribution of pixel intensity values in the image, and the histogram can visually display the grayscale distribution. Fig. 14 and

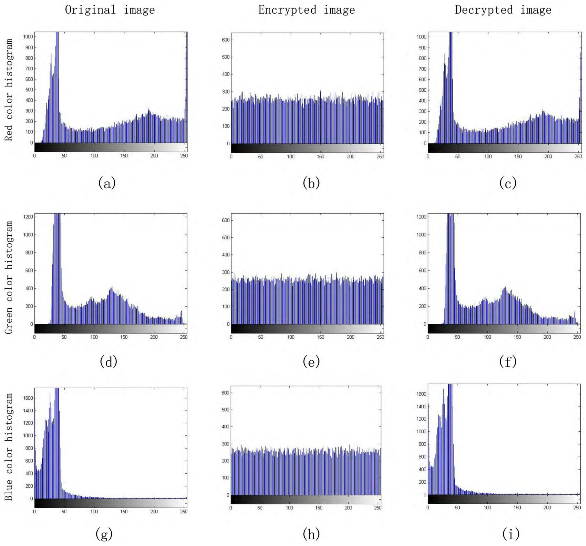


FIGURE 15. Histogram analysis: (a) (d) (g) are histograms of the three colors components of the Plain-image of "pepper", (b) (e) (h) are histograms of the three colors components of the cipher-image of "pepper", (c) (f) (i) are histograms of the three colors components of the decrypted image of "pepper".

Fig.15 show the histograms of the color components of the "sailboat" and "pepper" images. We can find that histograms of the cipher-images are very uniform. This means that it cannot provide any useful statistics in the cipher-image to trigger any statistical attacks on the algorithm.

In order to quantify and analyze the results of image histogram, we use the variance of the histograms to evaluate the uniformity of the encrypted images. The lower variance indicates the higher homogeneity of the encrypted image. The variance formula of the histogram is as follows [38].

$$var(z) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{1}{2}(z_i - z_j)^2 \tag{20}$$

where z denotes the gray or color level of the histogram value, M and N are the lengths and widths of the image respectively. and $\{z = (z_i = i | i = 1, 2, 3, \ldots, 256\}$. In this experiment, consider the sensitivity of the chaotic system to the initial value and parameter, we define a key subset p_1, p_2, φ_1, φ_2, ω_{01}, ω_{02}, ω_{03}, ω_{04}. In order to verify the stability of our algorithm for histogram uniformity with different keys, we set four group of key subsets:

$$\begin{cases} key1 = 0.16, 4.88, 0.22, 3.13, 0.27, 0.29, 0.8, 0.4 \\ key2 = 0.17, 4.87, 0.23, 3.14, 0.26, 0.3, 0.9, 0.5 \\ key3 = 0.18, 4.86, 0.24, 3.15, 0.25, 0.24, 0.6, 0.6 \\ key4 = 0.14, 4.9, 0.2, 3.11, 0.24, 0.23, 0.5, 0.7 \end{cases} \tag{21}$$

The variances of the encrypted images "sailboat" and ‘pepper‘ are obtained by $key1$, $key2$, $key3$, and $key4$ in equation (21) respectively. It is easy to see in Table 1 that the variances of the encrypted images are much lower than that of the plain-images. As shown in Fig.16:

TABLE 1. Variances between plain-images and cipher-images with different keys.

Encryption image	Plain-image	key	key1	key2	key3	key4
sailboat-red	53516	500.8163	507.6094	515.6599	489.5854	518.2562
pepper-red	110694	486.6938	509.8566	505.4953	484.5022	496.2009
sailboat-green	35773	480.5703	516.5623	530.5924	508.9600	499.4063
pepper-green	180719	507.2926	513.2553	494.2193	491.8458	507.0195
sailboat-blue	91207	526.7227	528.7341	540.6406	527.9521	501.2833
pepper-blue	431936	510.4410	485.6318	517.6523	490.9194	524.0567

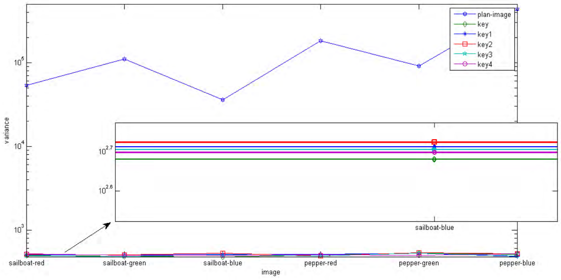


FIGURE 16. "sailboat" and "pepper" plain-images histogram variances and cipher-images histogram variances.

TABLE 2. Variances between plain-images and cipher-image with different keys.

Cipher-image	key(%)	key1(%)	key2(%)	key3(%)	key4(%)
sailboat-red	1.08	0.24	1.8	3.3	2.3
pepper-red	1.98	2.69	2.2	2.4	0.07
sailboat-green	5.2	1.8	4.6	0.34	1.5
pepper-green	0.9	2.1	1.7	2.1	0.85
sailboat-blue	0.32	0.7	2.9	0.55	4.5
pepper-blue	0.93	3.9	2.4	2.9	3.6

In order to analyze the stability of the histogram variance of different images encrypted by different keys, we define variance stability rate:

$$\eta_{var}(z) = \frac{|var - AVG_{var}|}{AVG_{var}} \times 100\% \tag{22}$$

where var is the variance of the image histogram. AVG_{var} is the average of the histogram variances using different keys to encrypt the corresponding image. From Table 2, we can see that our values of percentage are small. This experiment result indicates the variance of our algorithm is stable. Compared with reference [44], the average of η_{var} with five different secret keys is 3.97%, suggesting that our algorithm has excellent histogram uniformity and stability. It means that regardless of the difference in the variance of the histograms between different plain-images, this algorithm will hide the statistical properties of the cipher-image histogram. All the variances of the histograms are stable at about 500, so the proposed algorithm can effectively resist statistical attacks.

D. CORRELATION ANALYSIS

A plain-image often exhibits a certain degree of correlation between every two adjacent pixels. An effective encryption scheme can reduce the correlation between adjacent pixels. In order to get the correlation of two adjacent pixels, We have randomly selected 4000 pairs of adjacent pixels from plain-image and calculated the correlation coefficient as follows:

TABLE 3. Correlation coefficients of the plain-images and cipher-images.

Direction	Horizontal	Vertical	Diagonal
Plain sailboat	0.9577	0.9524	0.918
Plain pepper	0.9906	0.9847	0.9765
Encryption sailboat	-0.0076	0.0224	-0.0258
Encryption pepper	-0.0235	0.0003	0.0046
[2]	-0.0156	-0.0022	-0.0028
[36]	-0.0026	0.0041	0.0368
[45]	0.0090	0.0126	0.0069

$$\begin{aligned} r_{xy} &= \frac{cov(x, y)}{\sqrt{D(x)D(y)}} \\ E(x) &= \frac{1}{S}\sum_{i=1}^{S} x_i \\ D(x) &= \frac{1}{S}\sum_{i=1}^{S}(x_i - E(x))^2 \\ cov(x, y) &= \frac{1}{S}\sum_{i=1}^{S}(x_i - E(x))(y_i - E(y)) \end{aligned} \tag{23}$$

The x and y represent gray-level values of two adjacent pixels, and the correlation distribution is shown in Fig.17 and Fig.18. The proposed chaotic image segmentation method effectively reduces the correlation of pixels in that it is lower than the value in reference [2], [36], [45]. As seen in Table 3, the correlation coefficients are significantly reduced and approximately zero in the cipher-images. This further proves that our algorithm can effectively resist statistical attacks.

E. INFORMATION ENTROPY

The information entropy is used to evaluate the randomness of an image [46]. Its entropy value can be calculated by the following equation:

$$H(m) = -\sum_{i=0}^{2^{N-1}} p(m_i)log_2 p(m_i) \tag{24}$$

N is the number of bits to m_i, and $p(m_i)$ is the probability of m_i. For an 8-bit image, the ideal entropy of the random image is 8. We calculate the entropy of the RGB components of the encrypted image respectively. As can be seen from Table 4, our information entropy value is closer to 8 than that in reference [7], [47], [48]. It is proved that the proposed

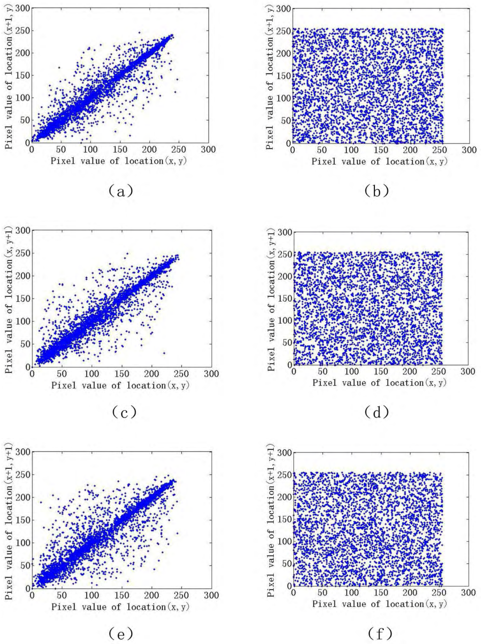


FIGURE 17. Pixel correlation coefficient analysis: (a) Horizontal adjacent pixel correlation of the red component of "sailboat" plain-image; (b) Horizontal adjacent pixel correlation of the red component of "sailboat" cipher-image; (c) Vertical adjacent pixel correlation of the green component of "sailboat" plain-image; (d) Vertical adjacent pixel correlation of the green component of "sailboat" cipher-image; (e) Diagonal adjacent pixel correlation of the blue component of "sailboat" plain-image; (f) Diagonal adjacent pixel correlation of the blue component of "sailboat" cipher-image.

TABLE 4. Information entropy for the encryption.

Cipher-image	Red	Green	Blue
our(sailboat)	7.9974	7.9971	7.9970
our(pepper)	7.9973	7.9976	7.9966
[7]	7.9899	7.9879	7.9883
[47]	7.9866	7.9852	7.9832
[48]	7.9851	7.9852	7.9852

algorithm can resist entropy attacks. Table 4. Information entropy for the encryption.

F. PEAK SIGNAL TO NOISE RATIO ANALYSIS

In this section, the difference between the plain-image and the cipher-image can be measured by mean square error (MSE). We utilize the Peak Signal-to-Noise Ratio (PSNR) to test the quality of the attacked cipher-image. It can be described as follows:

$$MSE = \frac{\sum_i \sum_j (P(i,j) - C(i,j))^2}{T} \times 100\%$$
$$PSNR = 10log_{10}(\frac{I_{max}^2}{MSE}) \tag{25}$$

where T represents the number of pixels in the image. $P(i,j)$ is the value of the pixels of plain-image; $C(i,j)$ is the pixel value of encrypted image; I_{max} is the maximum pixel value of the encrypted image. In a good encryption the scheme, PSNR should be as low as possible. The PSNR of the gray ''pepper'' image calculated in [2] is 8.9948, while the PSNR of our three colors components is lower. The MSE and PSNR values of the encrypted images are shown in Table 5.

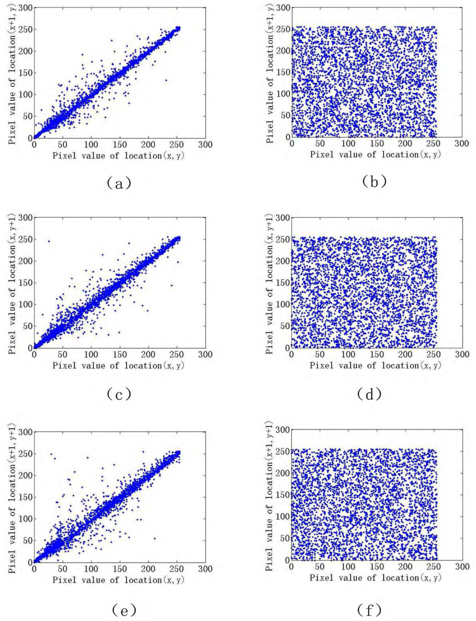


FIGURE 18. Pixel correlation coefficient analysis: (a) Horizontal adjacent pixel correlation of the red component of "pepper" plain-image; (b) Horizontal adjacent pixel correlation of the red component of "pepper" cipher-image; (c) Vertical adjacent pixel correlation of the green component of "pepper" plain-image; (d) Vertical adjacent pixel correlation of the green component of "pepper" cipher-image; (e) Diagonal adjacent pixel correlation of the blue component of "pepper" plain-image; (f) Diagonal adjacent pixel correlation of the blue component of "pepper" cipher-image.

TABLE 5. MSE and PSNR for the encryption.

Encryption image	Red	Green	Blue
MSE-sailboat	7243	11385	11450
PSNR-sailboat	9.5311	7.5674	7.5426
MSE-pepper	12303	10138	15089
PSNR-pepper	7.2308	8.0713	6.3442

G. DIFFERENTIAL ATTACK

In order to examine whether the proposed encryption algorithm can resist differential attacks. The number of pixels changing rate (NPCR) and the unified average changing intensity (UACI) [26] are two important evaluation factors for differential attacks analysis [36].

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%$$
$$UACI = \frac{1}{M \times N}[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}] \times 100\% \tag{26}$$

where $C_1(i,j)$ and $C_2(i,j)$ are the cipher-images before and after one pixel of the plain-image is changed, the $D(i,j)$ is defined by

$$D(i,j) = \begin{cases} 0, & C_1(i,j) = C_2(i,j) \\ 1, & C_1(i,j) \neq C_2(i,j) \end{cases} \tag{27}$$

TABLE 6. The NPCR values of the proposed algorithm.

Cipher-image	Red	Green	Blue
Our(sailboat)	99.5926	99.6262	99.6124
Our(pepper)	99.6231	99.6170	99.6124
[8]	99.60	99.59	99.58
[9]	98.8464	99.5880	99.6017
[11]	99.54	99.70	99.72
[50]	99.9192	99.7996	99.8015

TABLE 7. The UACI values of the proposed algorithm.

Cipher-image	Red	Green	Blue
Our(sailboat)	33.4792	33.4738	33.4728
Our(pepper)	33.4723	33.4649	33.4683
[8]	33.44	33.47	33.39
[9]	31.9382	33.9716	35.3876
[11]	33.37	33.42	33.40
[50]	33.4707	33.4826	33.7332

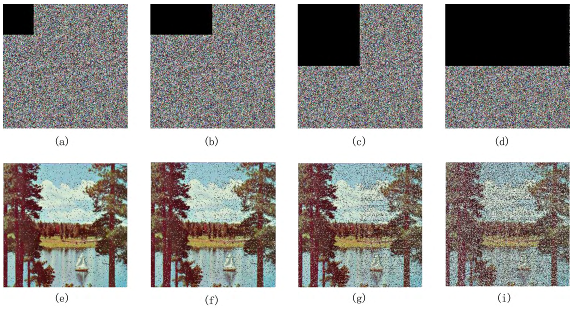


FIGURE 19. Cropping attack analysis: (a) 1/16 cropping attack, (b) 1/8 cropping attack, (c)1/4 cropping attack, (d)1/2 cropping attack, (e) decrypted image of (a), (f) decrypted image of (b), (g) decrypted image of (c), (i) decrypted image of (d).

NPCR and UACI of "Sailboat" and "pepper" are shown in Table 6 and Table 7. NPCR is close to the theoretical value of 99.6094% and the UACI is close to the theoretical value of 33.4653% [49], which means that our scheme can resist differential attacks. Meanwhile, our algorithm is superior to the literature [8], [9], [11], [50].

H. CROPPING ATTACK

An ideal cryptosystem should against data loss attacks by transmission and storage [51]. To evaluate its robustness of resisting cropping attacks, parts with 64 × 64, 64 × 128, 128 × 128, 128 × 256 are deleted from the cipher-image "sailboat" as shown in Fig 19 (a)-(d). The decryption images are shown in Fig 19 (e)-(h), and they can still be recognized. It proves that our algorithm has the ability to resist data cropping attacks.

I. NOISE ATTACK

A good encryption algorithm should be able to resist noise attacks. Attacks with 0.01, 0.05, 0.1 salt and pepper noise are performed as shown in Fig.20. It can be seen that the decryption images after adding 0.1 salt and pepper noise are still identifiable. Therefore, our algorithm has good robustness and can efficiently resist noise attacks.

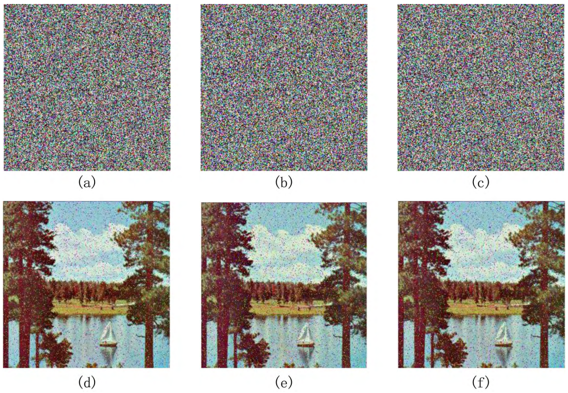


FIGURE 20. Noise attack analysis: (a) adding 0.01 salt and pepper noise, (b) adding 0.05 salt and pepper noise, (c) adding 0.1 salt and pepper noise, (d) decrypted image of (a), (e) decrypted image of (b), (f) decrypted image of (c).

TABLE 8. Speed analysis.

Algorithm	Entropy(speed)
Ours	7.2428
[10]	9.89
[13]	41.1227
[52]	8.11
[53]	9.2115

J. SPEED TEST

Apart from the security considerations, some other aspects of the image cryptosystem algorithm are also important, particularly the running speed for real-time Internet multimedia applications. The computer configuration is 2.5 GHZ CPU, 8 GB of memory and Microsoft Windows 10 operating system. We utilize MATLAB 2014 to simulate the encryption operations. This encryption algorithm generates chaotic key pool by QCNN, which reduces the number of iterations of high-dimensional chaotic map QCNN. Because of the existence of chaotic key pool, there is no need to iterate the chaotic system repeatedly, and the performance of the proposed algorithm will be improved with the increase of the amount of data. Compared with other similar color image encryption schemes [10], [13], [52], [53] in Table 8, the speed of our algorithm shows some advantages. At the same time, due to the symmetric structure, the time cost of encryption and decryption are the same. Therefore, the proposed algorithm shows that it can communicate in real time.

V. CONCLUSION

In this paper, a novel image segmentation encryption algorithm with a fast chaotic key generation scheme is proposed. In order to destroy pixel correlations, the plain-image is divided into two blocks by a chaotic sequence, and they are scrambled through intra-block and inter-block. Additionally, one group of chaotic sequence is regarded as a key pool, which is calculated by iterating a QCNN before encryption. The key is selected from the pool as an index of segmentation, scrambling and diffusion. This key pool scheme reduces chaotic iteration times, so it improves keys generation efficiency. Experiments and performance analysis indicate that our algorithm has well-security and good performance.

REFERENCES

[1] F. A. Abdullatif, A. A. Abdullatif, and A. Al-Saffar, "Hiding techniques for dynamic encryption text based on corner point," *J. Phys., Conf. Ser.*, vol. 1003, no. 1, 2018, Art. no. 012027.

[2] Q. Yin and C. Wang, "A new chaotic image encryption scheme using breadth-first search and dynamic diffusion," *Int. J. Bifurcation Chaos*, vol. 28, no. 4, May 2018, Art. no. 1850047.

[3] N. K. Sreelaja and N. K. Sreeja, "An image edge based approach for image password encryption," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5733–5745, 2016.

[4] Z. E. Dawahdeh, S. N. Yaakob, and R. R. B. Othman, "A new image encryption technique combining elliptic curve cryptosystem with hill cipher," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 30, no. 3, pp. 349–355, Jul. 2018.

[5] T. Zhao, Q. Ran, L. Yuan, Y. Chi, and J. Ma, "Security of image encryption scheme based on multi-parameter fractional Fourier transform," *Opt. Commun.*, vol. 376, pp. 47–51, Oct. 2016.

[6] X. Wang, L. Teng, and X. Qin, "A novel colour image encryption algorithm based on chaos," *Signal Process.*, vol. 92, no. 4, pp. 1101–1108, Apr. 2012.

[7] X. Wu, C. Bai, and H. Kan, "A new color image cryptosystem via hyperchaos synchronization," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 6, pp. 1884–1897, 2014.

[8] P. Praveenkumar, R. Amirtharajan, K. Thenmozhi, and J. B. B. Rayappan, "Triple chaotic image scrambling on RGB—A random image encryption approach," *Secur. Commun. Netw.*, vol. 8, no. 18, pp. 3335–3345, 2015.

[9] X.-L. Chai, Z.-H. Gan, Y. Lu, M.-H. Zhang, and Y.-R. Chen, "A novel color image encryption algorithm based on genetic recombination and the four-dimensional memristive hyperchaotic system," *Chin. Phys. B*, vol. 25, no. 10, 2016, Art. no. 100503.

[10] C. Jin and H. Liu, "A color image encryption scheme based on arnold scrambling and quantum chaotic," *IJ Netw. Secur.*, vol. 19, no. 3, pp. 347–357, 2017.

[11] X.-Y. Wang, P. Li, Y.-Q. Zhang, L.-Y. Liu, H. Zhang, and X. Wang, "A novel color image encryption scheme using DNA permutation based on the Lorenz system," *Multimedia Tools Appl.*, vol. 77, no. 5, pp. 6243–6265, 2018.

[12] X. Wu, B. Zhu, Y. Hu, and Y. Ran, "A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps," *IEEE Access*, vol. 5, pp. 6429–6436, 2017.

[13] Y. Zhou, L. Bao, and C. P. Chen, "A new 1D chaotic system for image encryption," *Signal Process.*, vol. 97, no. 11, pp. 172–182, 2014.

[14] R. Matthews, "ON The derivation of a 'chaotic' encryption algorithm," *Cryptologia*, vol. 13, no. 1, pp. 29–42, 1989.

[15] X. Tong and M. Cui, "Image encryption with compound chaotic sequence cipher shifting dynamically," *Image Vis. Comput.*, vol. 26, no. 6, pp. 843–850, 2008.

[16] K.-W. Wong, B. S.-H. Kwok, and C.-H. Yuen, "An efficient diffusion approach for chaos-based image encryption," *Chaos, Solitons Fractals*, vol. 41, pp. 2652–2663, Sep. 2009.

[17] A. Kumar and M. K. Ghose, "Extended substitution–diffusion based image cipher using chaotic standard map," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 16, no. 1, pp. 372–382, 2010.

[18] G. Zhang and Q. Liu, "A novel image encryption method based on total shuffling scheme," *Opt. Commun.*, vol. 284, no. 12, pp. 2775–2780, 2011.

[19] C. Zhu, "A novel image encryption scheme based on improved hyperchaotic sequences," *Opt. Commun.*, vol. 285, no. 1, pp. 29–37, 2012.

[20] Y.-Q. Zhang, Y. He, and X.-Y. Wang, "Spatiotemporal chaos in mixed linear–nonlinear two-dimensional coupled logistic map lattice," *Phys. A, Stat. Mech. Appl.*, vol. 490, pp. 148–160, Jan. 2018.

[21] Y. Zhang, X. Wang, L. Liu, and J. Liu, "Fractional order spatiotemporal chaos with delay in spatial nonlinear coupling," *Int. J. Bifurcation Chaos*, vol. 28, no. 2, 2018, Art. no. 1850020.

[22] Y.-Q. Zhang, X.-Y. Wang, L.-Y. Liu, Y. He, and J. Liu, "Spatiotemporal chaos of fractional order logistic equation in nonlinear coupled lattices," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 52, pp. 52–61, Nov. 2017.

[23] Y.-Q. Zhang, X.-Y. Wang, J. Liu, and Z.-L. Chi, "An image encryption scheme based on the MLNCML system using DNA sequences," *Opt. Lasers Eng.*, vol. 82, pp. 95–103, Jul. 2016.

[24] Y.-Q. Zhang and X.-Y. Wang, "Spatiotemporal chaos in mixed linear–nonlinear coupled logistic map lattice," *Phys. A, Stat. Mech. Appl.*, vol. 402, pp. 104–118, May 2014.

[25] Y. Mao, G. Chen, and S. Lian, "A novel fast image encryption scheme based on 3D chaotic baker maps," *Int. J. Bifurcation Chaos*, vol. 14, no. 10, pp. 3613–3624, Oct. 2004.

[26] S. M. Seyedzadeh, B. Norouzi, M. R. Mosavi, and S. Mirzakuchaki, "A novel color image encryption algorithm based on spatial permutation and quantum chaotic map," *Nonlinear Dyn.*, vol. 81, nos. 1–2, pp. 511–529, 2015.

[27] G. Ye and J. Zhou, "A block chaotic image encryption scheme based on self-adaptive modelling," *Appl. Soft Comput.*, vol. 22, no. 5, pp. 351–357, 2014.

[28] X. Wang, L. Liu, and Y. Zhang, "A novel chaotic block image encryption algorithm based on dynamic random growth technique," *Opt. Lasers Eng.*, vol. 66, pp. 10–18, Mar. 2015.

[29] L. Xu, X. Gou, Z. Li, and J. Li, "A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion," *Opt. Lasers Eng.*, vol. 91, pp. 41–52, Apr. 2017.

[30] G. Toth, C. S. Lent, P. D. Tougaw, Y. Brazhnik, W. Weng, W. Porod, R.-W. Liu, and Y.-F. Huang, "Quantum cellular neural networks," *Superlattices Microstructures*, vol. 20, no. 4, pp. 473–478, 1996.

[31] L. Fortuna and D. Porto, "Quantum-CNN to generate nanoscale chaotic oscillators," *Int. J. Bifurcation Chaos*, vol. 14, no. 3, pp. 1085–1089, 2004.

[32] I. Amlani, A. O. Orlov, G. Toth, G. H. Bernstein, C. S. Lent, and G. L. Snider, "Digital logic gate using quantum-dot cellular automata," *Science*, vol. 284, no. 5412, pp. 289–291, 1999.

[33] L. Fortuna and D. Porto, "Chaotic phenomena in quantum cellular neural networks," in *Proc. 7th IEEE Int. Workshop Cellular Neural Netw. Appl.*, Jul. 2002, pp. 369–376.

[34] W. Sen, C. Li, L. Qin, and W. Gang, "Chaotic phenomena in Josephson circuits coupled quantum cellular neural networks," *Chin. Phys.*, vol. 16, no. 9, p. 2631, 2007.

[35] W. Sen, C. Li, K. Qiang, W. Gang, and L. Qin, "The characteristics of nonlinear chaotic dynamics in quantum cellular neural networks," *Chin. Phys. B*, vol. 17, no. 8, p. 2837, 2008.

[36] X. Di, J. Li, H. Qi, L. Cong, and H. Yang, "A semi-symmetric image encryption scheme based on the function projective synchronization of two hyperchaotic systems," *PLoS ONE*, vol. 12, no. 9, 2017, Art. no. e0184586.

[37] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernstein, "Quantum cellular automata," *Nanotechnology*, vol. 4, no. 1, p. 49, 1993.

[38] W. Porod, "Quantum-dot devices and quantum-dot cellular automata," *Int. J. Bifurcation Chaos*, vol. 07, no. 10, pp. 2199–2218, 1997.

[39] W. Hao-Xiang, C. Guo-Liang, M. Sheng, and T. Li-Xin, "Nonlinear feedback control of a novel hyperchaotic system and its circuit implementation," *Chin. Phys. B*, vol. 19, no. 3, 2010, Art. no. 030509.

[40] T. Gao and Z. Chen, "A new image encryption algorithm based on hyper-chaos," *Phys. Lett. A*, vol. 372, no. 4, pp. 394–400, 2008.

[41] H. Q. ling, S.-N. Xue, Y.-Y. Deng, Q. Xu, and H.-J. Wang, "A new chaotic system and its linear feedback synchronization," *J. Chengdu Univ. Inf. Eng.*, vol. 32, no. 5, pp. 27–32, 2017.

[42] S. M. Seyedzadeh and S. Mirzakuchaki, "A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map," *Signal Process.*, vol. 92, no. 5, pp. 1202–1215, May 2012.

[43] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. Bifurcation Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.

[44] Y.-Q. Zhang and X.-Y. Wang, "A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice," *Inf. Sci.*, vol. 273, no. 8, pp. 329–351, 2014.

[45] I. A. Ismail, M. Amin, and H. Diab, "A digital image encryption algorithm based a composition of two chaotic logistic maps," *IJ Netw. Secur.*, vol. 11, no. 1, pp. 1–10, 2010.

[46] X. Wang, X. Zhu, and Y. Zhang, "An image encryption algorithm based on Josephus traversing and mixed chaotic map," *IEEE Access*, vol. 6, no. 1, pp. 23733–23746, 2018.

[47] H. Liu and X. Wang, "Color image encryption based on one-time keys and robust chaotic maps," *Comput. Math. Appl.*, vol. 59, no. 10, pp. 3320–3327, 2010.

[48] H. Liu, A. Kadir, and X. Sun, "Chaos-based fast colour image encryption scheme with true random number keys from environmental noise," *IET Image Process.*, vol. 11, no. 5, pp. 324–332, 2017.

[49] G. Ye, C. Pan, X. Huang, Z. Zhao, and J. He, "A chaotic image encryption algorithm based on information entropy," *Int. J. Bifurcation Chaos*, vol. 28, no. 1, 2018, Art. no. 1850010.

[50] B. Norouzi and S. Mirzakuchaki, "A fast color image encryption algorithm based on hyper-chaotic systems," *Nonlinear Dyn.*, vol. 78, no. 2, pp. 995–1015, Oct. 2014.

[51] X. Li, C. Zhou, and N. Xu, "A secure and efficient image encryption algorithm based on DNA coding and spatiotemporal chaos," *IJ Netw. Secur.*, vol. 20, no. 1, pp. 110–120, 2018.

[52] M. Ghebleh, A. Kanso, and H. Noura, "An image encryption scheme based on irregularly decimated chaotic maps," *Signal Process. Image Commun.*, vol. 29, no. 5, pp. 618–627, 2014.

[53] S. Mazloom and A. M. Eftekhari-Moghadam, "Color image encryption based on coupled nonlinear chaotic map," *Chaos, Solitons Fract.*, vol. 42, no. 3, pp. 1745–1754, 2009.

ZHENLONG MAN received the B.S. degree in science from the Jilin Institute of Chemical Technology, in 2017. He is currently pursuing the Ph.D. degree in computer science and technology with the Changchun University of Science and Technology, China. His research interests include information security, chaotic system, and image security.

JINQING LI received the B.S. degree from the Changchun University of Technology, in 2002, and the M.S. and Ph.D. degrees from the Changchun University of Science and Technology, in 2007 and 2014, respectively, where she is currently an Associate Professor. She was a Visiting Scholar with the Florida International University, from 2018 to 2019. Her major research interests include cybersecurity, information security, and chaotic encryption.

XIAOQIANG DI received the B.S. degree in computer science and technology from the Changchun University of Science and Technology, in 2002, and the M.S. and Ph.D. degrees in communication and information systems from the Changchun University of Science and Technology, in 2007 and 2014, respectively. He was a Visiting Scholar with the Norwegian University of Science and Technology, Norway, from 2012 to 2013. He is currently a Professor and the Ph.D. Supervisor with the Changchun University of Science and Technology. His major research interests include network information security and integrated network.

OU BAI (M'07) received the B.S. degree in electronic engineering from Tsinghua University, China, and the Ph.D. degree in advanced systems control engineering from Saga University, Japan. He received the postdoctoral research training with the National Institutes of Health. He currently serves as the Director of the Human Cyber-Physical Systems Laboratory, Department of Electrical and Computer Engineering, Florida International University. He has published 100 peer-reviewed journal papers, book chapters, and conference proceeding papers. His research is highly interdisciplinary with collaborations from academia, industry, medical institutes, and government laboratories. His research has been well-supported by the industry and federal agents, including the National Institutes of Health and National Science Foundation (IEEE Number: 90599290).