International Journal of Bifurcation and Chaos, Vol. 29, No. 9 (2019) 1950115 (17 pages)

DOI: 10.1142/S0218127419501153

A Novel Color Image Encryption Algorithm Based on Hyperchaotic System and Permutation-Diffusion Architecture

Guangfeng Cheng*, Chunhua Wang† and Hua Chen‡
College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, P. R. China
*gfcheng@hnu.edu.cn
†wch1227164@hnu.edu.cn
‡anneychen@126.com

Received March 3, 2018; Revised November 28, 2018

In recent years, scholars studied and proposed some secure color image encryption algorithms. However, the majority of the published algorithms encrypted red, green and blue (called R, G, B for short) components independently. In the paper, we propose a color image encryption scheme based on hyperchaotic system and permutation-diffusion architecture. The encryption algorithm utilizes a block permutation which is realized by mixing R, G, B components to strengthen the dependence of each component. Besides, it can reduce time consumption. Then, the key streams generated by the hyperchaotic system are exploited to diffuse the pixels, the three components affect each other again. And in the diffusion process, we can get two totally different encrypted images even though we change the last pixel because the G component is diffused in reverse order. The experimental results reveal that our algorithm possesses better abilities of resisting statistical attacks and differential attacks, larger key space, closer information entropy to 8, and faster encryption speed compared with other chaos-based color image encryption algorithms.

Keywords: Hyperchaotic system; multiwing; color image encryption; block permutation.

1. Introduction

With the quick development of network and communication techniques, the security of image information has attracted more and more attention. However, due to a variety of intrinsic characteristics of images such as a strong correlation of adjacent pixels, data redundancy, and high computation complexity, the traditional encryption algorithms like MD5, IDEA, AES, etc., are not suited to encrypt image. Chaos-based encryption algorithms are more suitable than traditional encryption algorithms [Baptista, 1998; Jakimoski & Kocarev, 2001; Schmitz, 2001], because chaotic systems or maps possess some excellent characteristics such as randomness, ergodicity, sensitivity to initial conditions and system parameters [Zhou *et al.*, 2018a; Zhang & Wang, 2019; Zhou *et al.*, 2018b]. Hence, chaotic image encryption has become more and more popular.

In recent years, scholars have proposed a lot of chaotic image encryption algorithms. For example, some algorithms possess larger key and higher security. In [Liu & Wang, 2010], a stream-cipher algorithm based on one-time keys and robust chaotic maps was designed to get high security and improve the dynamical degradation. Sui *et al.* [Sui & Gao,

†Author for correspondence

2013] proposed a color image encryption algorithm based on gyrator transform and Arnold transform. In [Bakhshandeh & Eslami, 2013], a novel image encryption algorithm using chaotic maps, permutation-diffusion architecture and cellular automata was introduced to improve secure diffusion mechanism and computational efficiency. In [Belazi *et al.*, 2016a], a novel image encryption approach was proposed. The scheme consists of diffusion, substitution, diffusion and permutation. In [Pak & Huang, 2017], a color image encryption scheme based on 1D chaotic map was proposed. It generated a chaotic system by using two same 1D chaotic maps. Meanwhile, in order to reduce the strong correlation of adjacent pixels, some researchers proposed a number of image encryption schemes using 2D Arnold cat map [Li *et al.*, 2017a; Som *et al.*, 2015; Ye *et al.*, 2015]. Chen *et al.* [2009] proposed a color image encryption scheme using interference technique and Arnold transform. Liu *et al.* [2016] proposed an image encryption scheme using Baker map of varying parameters to overcome weaknesses that the chaotic orbits, their parameters and initial conditions may estimate. In [Tong *et al.*, 2016a], a joint image encryption algorithm using lossless compression and chaotic map was proposed. The algorithm employed SPIHT (Set Partitioning in Hierarchical Trees) encoding method and many kinds of chaotic maps. However, these encryption algorithms using the low-dimensional chaotic maps have a few weaknesses such as poor efficiency, smaller key space and weaker security. Because of the better characteristics of high-dimensional systems, some researchers proposed a number of encryption algorithms using high-dimensional chaotic systems in recent years [Amin, 2015; Belazi *et al.*, 2016b; Wang *et al.*, 2016a; Xie *et al.*, 2016; Mollaeefar *et al.*, 2015; Norouzi & Mirzakuchaki, 2014; Seyedzadeh *et al.*, 2015; Wang *et al.*, 2012; Li *et al.*, 2017b; Wang *et al.*, 2016b; Yuan *et al.*, 2016; Zhang *et al.*, 2016]. In [Liu *et al.*, 2016; Mao *et al.*, 2004; Tong & Cui, 2009], several encryption schemes were proposed by extending low-dimensional maps to corresponding high-dimensional maps. Gao *et al.* [Gao & Chen, 2008] proposed a hyperchaos-based image encryption algorithm. Key streams of the algorithm generated by a chaotic system are independent of the plaintext and diffusion process. Therefore, it cannot withstand chosen-plaintext attack and chosen-ciphertext attack. An image encryption algorithm based on simple perceptron and high-dimensional chaotic system was presented in [Wang *et al.*, 2010]. In [Liu & Wang, 2011], the authors proposed a bit-level permutation and high-dimension chaotic map to encrypt color image. Firstly, they converted the plain color image of size $(M \times N)$ into a grayscale image of size $(M \times 3N)$, then transformed it into a binary matrix, and permuted the matrix at bit-level by the scrambling mapping generated by piecewise linear chaotic map (PWLCM). Secondly, the Chen system is used to confuse and diffuse the red, green and blue components simultaneously. Som *et al.* [2015] proposed a color image encryption algorithm based on multiple chaotic maps. The encryption algorithm utilized Arnold cat map to permute the original image. After that, the intermediate image was encrypted by chaotic sequences that were generated by multiple chaotic maps. To improve efficiency and security, Ye *et al.* [2015] proposed a chaotic image encryption algorithm using SHA-3 hash function and double Arnold maps. To reduce time consumption in permutation stage, the plain-image was permutated in four different directions by a novel wave-line. Jawad *et al.* [Jawad & Sulong, 2015] proposed a security-enhanced image encryption scheme using four-dimensional hyperchaotic system and XOR operator to enhance the security. Norouzi *et al.* [Norouzi & Mirzakuchaki, 2014] proposed a hyperchaos-based color image encryption scheme. However, the security was not good due to their inappropriate cryptographic structure. Therefore, Tong *et al.* [2016b] proposed an enhanced algorithm to overcome the weakness. In [Zhu *et al.*, 2017], a novel image encryption algorithm using two-dimensional compound homogeneous hyperchaotic system (CHHCS) and local binary pattern was proposed. In [Wang *et al.*, 2015b; Hu *et al.*, 2017], two encryption algorithms using DNA sequence operations were proposed. They employed DNA encoding to encrypt images. In [Liu *et al.*, 2012], a novel confusion and diffusion method for image encryption algorithm based on PWLCM system and DNA coding was proposed. In [Hua & Zhou, 2017; Xu *et al.*, 2017], two encryption schemes using block scrambling were proposed. In [Hua & Zhou, 2017], an image encryption algorithm using block-based scrambling and image filtering was proposed. The block-based permutation can disperse adjacent pixels, and it can reduce the high correlations between adjacent pixels. In [Xu *et al.*, 2017], the authors proposed a

chaotic image encryption scheme based on block scrambling and dynamic index. In [Huang & Ye, 2014], an efficient self-adaptive model for chaotic image encryption algorithm was proposed to solve the problem of fixed chaotic sequence produced by the same initial conditions but for different images. However, a defect still exists in the aforementioned color image encryption algorithms. This is because the red, green and blue components are encrypted independently. If the gray image is cracked successfully by the adversary, then the color image will also be cracked soon.

To solve the problem, we propose a color image encryption scheme based on hyperchaotic system and permutation-diffusion architecture. The encryption algorithm utilizes a block permutation which is realized by mixing R, G and B components to strengthen the dependence of each component. And time of iterating the hyperchaotic system will be greatly reduced. After permutation, each component has a half of pixels of other components. Then, the chaotic sequences generated by a hyperchaotic system are exploited to diffuse the pixels, the three components affect each other again. And in the diffusion process, we diffuse the G component in reverse order. Because the R, G and B three components affect each other in the permutation and diffusion of color image encryption algorithms and key stream generation, our algorithm has better abilities of resisting statistical attacks and differential attacks, larger key space, closer information entropy to 8, and faster encryption speed compared with other chaos-based color image encryption algorithms. In addition, it is a failure that the adversary uses the gray image to crack our algorithm, because the proposed scheme can mix R, G and B components to strengthen the dependence of each component, which is more efficient than former color image encryption algorithms in resisting gray image attack.

We organize the rest of the paper as follows. In Sec. 2, the novel color image encryption scheme is introduced in detail. In Sec. 3, experimental results and performance analysis are shown [Özkaynak, 2018]. Finally, the conclusions are given in Sec. 4.

2. Proposed Cryptosystem

The flow diagram of the encryption algorithm is given in Fig. 1. Firstly, divide the original color image into three components red, green and blue. Secondly, the initial conditions of the chaotic system are produced by the plain-image. Thirdly, to reduce time consumption of sorting operation at the permutation stage, a block permutation scheme is suggested. In the permutation process, the R, G and B images are divided into nonoverlapping blocks of equal size. Then, we exchange pixels of R with G and B according to S^1, S^2 and S^3 (produced by chaotic system in Sec. 2.2). Therefore, the R component possesses a half of pixels of G and a half of pixels of B. Next, we swap pixels of G and B according to S^4 and S^5 (produced by the chaotic system in Sec. 2.2). After that, each component has a half of pixels of the other components.

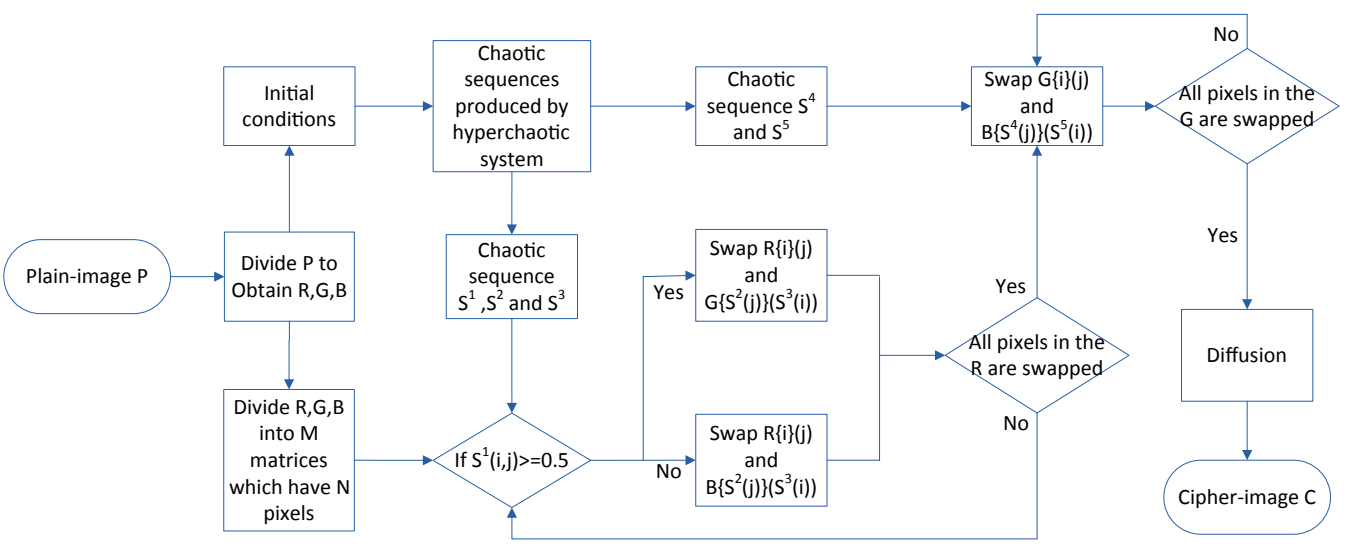


Fig. 1. Flow diagram of the proposed algorithm.

The block permutation not only reduces time consumption but also enhances the dependence of R, G and B. Finally, a diffusion process is employed. Then, we can obtain the encrypted image C by combining CR, CG and CB.

2.1. *Hyperchaotic system*

The algorithm adopts a 5D multiwing hyperchaotic system as follows [Amin, 2015]:

$$\begin{cases} \dot{x}_1 = -ax_1 + x_2x_3, \\ \dot{x}_2 = -bx_2 + fx_5, \\ \dot{x}_3 = -cx_3 + gx_4 + x_1x_2, \\ \dot{x}_4 = dx_4 - hx_1, \\ \dot{x}_5 = ex_5 - x_2x_1^2, \end{cases} \tag{1}$$

where x_1, x_2, x_3, x_4, x_5 are state variables and a, b, c, d, e, f, g, h are real constant parameters of the chaotic system (1). When $a = 10$, $b = 60$, $c = 20$, $d = 15$, $e = 40$, $f = 1$, $g = 50$, $h = 10$, the system is hyperchaotic.

2.2. *Generation of the initial conditions and scrambling sequences*

We suppose the size of the original color image P is $M \times N$. The initial conditions and scrambling sequences are generated by the following steps:

Step 1. We get three matrices R, G and B by dividing the plain-image P into red, green and blue components. Then the initial conditions are generated by the following formulae (2)–(6).

$$x_1 = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} R(i,j) + \sum_{i=1}^{M/2}\sum_{j=1}^{N} G(i,j) + \sum_{i=M/2+1}^{M}\sum_{j=1}^{N} B(i,j)}{2 * M * N * 2^8}, \tag{2}$$

$$x_2 = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} R(i,j) + \sum_{i=M/2+1}^{M}\sum_{j=1}^{N} G(i,j) + \sum_{i=1}^{M/2}\sum_{j=1}^{N} B(i,j)}{2 * M * N * 2^8}, \tag{3}$$

$$x_3 = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} R(i,j) + \sum_{i=1}^{M}\sum_{j=1}^{N/2} G(i,j) + \sum_{i=1}^{M}\sum_{j=N/2+1}^{N} B(i,j)}{2 * M * N * 2^8}, \tag{4}$$

$$x_4 = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} R(i,j) + \sum_{i=1}^{M}\sum_{j=N/2+1}^{N} G(i,j) + \sum_{i=1}^{M}\sum_{j=1}^{N/2} B(i,j)}{2 * M * N * 2^8}, \tag{5}$$

$$x_5 = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} G(i,j) + \sum_{i=1}^{M}\sum_{j=1}^{N} B(i,j)}{2 * M * N * 2^8}, \tag{6}$$

where $R(i,j)$ represents the pixel in the ith row and jth column of the red component. We can get initial conditions x_1, x_2, x_3, x_4 by using formulae (2)–(5). In formula (6), we use $M \times N$ pixels of green and $M \times N$ pixels of blue to get the initial condition x_5.

Step 2. Utilize the above initial conditions x_1, x_2, x_3, x_4, x_5 to iterate the hyperchaotic system (1) for $t + \lceil MN/5 \rceil$ ($t \geq 1000$) times. Then abandon the former t values to eliminate the transient effect. Next, we will obtain a chaotic sequence with length MN: $S = \{s_1, s_2, \ldots, s_{M\times N}\}$.

Step 3. Obtain the sequence S^1 according to the formula (7).

$$S_i^1 = \mathrm{mod}(S_i \times 10^4, 1) \quad i \in (1, 2, \ldots, M \times N). \tag{7}$$

Step 4. According to the formulae (8)–(11), we can get sequences S^{12}, S^{13}, S^{14} and S^{15}. And each value in sequences is 0 to M.

$$S_i^{12} = \mathrm{mod}(S_{1000+i}^1 \times 10^{14}, M) \quad i \in (1, 2, \ldots, 1000), \tag{8}$$

$$S_i^{13} = \mathrm{mod}(S_{2000+i}^1 \times 10^{14}, M) \quad i \in (1, 2, \ldots, 1000), \tag{9}$$

$$S_i^{14} = \mathrm{mod}(S_{3000+i}^1 \times 10^{14}, M) \quad i \in (1, 2, \ldots, 1000), \tag{10}$$

$$S_i^{15} = \mathrm{mod}(S_{4000+i}^1 \times 10^{14}, M) \quad i \in (1, 2, \ldots, 1000). \tag{11}$$

Step 5. We choose M values for which each value is different from the others to obtain sequences S^2, S^3, S^4 and S^5 from sequences S^{12}, S^{13}, S^{14}, S^{15}.

2.3. *Proposed encryption algorithm*

Step 1. The R, G and B images are divided into nonoverlapping blocks of equal size, i.e. M blocks which have N pixels. Then we convert M blocks digital image matrix to one-dimensional vectors. And we can get $3M$ one-dimensional vectors which have N pixels.

Step 2. Obtain the sequences S^{12}, S^{13}, S^{14} and S^{15} by iterating the hyperchaotic system and process the chaotic sequences S, as described in Sec. 2.2.

Step 3. At the permutation stage, we exchange pixels of R with G and B according to the chaotic sequences S^1, S^2 and S^3 by the following formulae (12) and (13). Therefore, the R component possesses a half of pixels of G and a half of pixels of B.

$$\begin{cases} IR\{i\}(j) = G\{S^2(j)\}(S^3(i)) \\ IG\{S^2(j)\}(S^3(i)) = R\{i\}(j) \end{cases} \quad \text{if } S^1((i-1)*N+j) > 0.5, \tag{12}$$

$$\begin{cases} IR\{i\}(j) = B\{S^2(j)\}(S^3(i)) \\ IB\{S^2(j)\}(S^3(i)) = R\{i\}(j) \end{cases} \quad \text{if } S^1((i-1)*N+j) \le 0.5, \tag{13}$$

where $i = 1, 2, \ldots, M, j = 1, 2, \ldots, N$ and $IR\{i\}(j)$ represent the jth pixel of the ith block.

Step 4. Because the other half of G and B is not permuted, we exchange pixels of G with B according to the chaotic sequences S^4 and S^5 by the following formula (14). After that, we can obtain scrambling images IR, IG and IB. And each component has a half of pixels of other components. Besides, the permutation process also reduces time consumption.

Step 5. Modify the original initial conditions by the following formulae (15)–(19). Then, iterate the hyperchaotic system (1) for $t + \lceil MN/5 \rceil$ $(t \ge 1000)$ times. Then abandon the former t values to eliminate the transient effect. Next, we will obtain three chaotic sequences with length MN: $K^1 = \{k_1^1, k_2^1, \ldots, k_{MN}^1\}$, $K^2 = \{k_1^2, k_2^2, \ldots, k_{MN}^2\}$, $K^3 = \{k_1^3, k_2^3, \ldots, k_{MN}^3\}$.

$$\begin{cases} \mathrm{temp} = IG\{i\}(j), \\ IG\{i\}(j) = IB\{S^4(j)\}(S^5(i)), \\ IB\{S^4(j)\}(S^5(i)) = \mathrm{temp}, \end{cases} \tag{14}$$

$$x_1' = \mathrm{mod}\left(\frac{x_2 + x_3 + x_4 + x_5}{4} * 10^4, 1\right), \tag{15}$$

$$x_2' = \mathrm{mod}\left(\frac{x_1 + x_3 + x_4 + x_5}{4} * 10^4, 1\right), \tag{16}$$

$$x_3' = \mathrm{mod}\left(\frac{x_1 + x_2 + x_4 + x_5}{4} * 10^4, 1\right), \tag{17}$$

$$x_4' = \mathrm{mod}\left(\frac{x_1 + x_2 + x_3 + x_5}{4} * 10^4, 1\right), \tag{18}$$

$$x_5' = \mathrm{mod}\left(\frac{x_1 + x_2 + x_3 + x_4}{4} * 10^4, 1\right). \tag{19}$$

Step 6. Obtain the key streams $K^4 = \{k_1^4, k_2^4, \ldots, k_{M\times N}^4\}$, $K^5 = \{k_1^5, k_2^5, \ldots, k_{M\times N}^5\}$, $K^6 = \{k_1^6, k_2^6, \ldots, k_{M\times N}^6\}$ by processing the chaotic sequences K^1, K^2, K^3 according to the following formulae (20)–(22).

$$K_i^4 = \mathrm{mod}((|K_i^1| - \lfloor |K_i^1| \rfloor) * 10^{14}, 256) \quad i = 1, 2, \ldots, M \times N, \tag{20}$$

$$K_i^5 = \mathrm{mod}((|K_i^2| - \lfloor |K_i^2| \rfloor) * 10^{14}, 256) \quad i = 1, 2, \ldots, M \times N, \tag{21}$$

$$K_i^6 = \mathrm{mod}((|K_i^3| - \lfloor |K_i^3| \rfloor) * 10^{14}, 256) \quad i = 1, 2, \ldots, M \times N. \tag{22}$$

Step 7. Convert digital image matrix IR, IG and IB to one-dimensional vectors and diffuse the scrambling images to get encrypted images CR, CG and CB by the following formulae (23)–(25).

$$CR_i = \mathrm{mod}(IR_i + CR_{i-1}, 256) \oplus K_i^4 \oplus \mathrm{mod}(CG_{M\times N-i+1} + CB_{i-1}, 256), \quad (23)$$

$$CG_{M\times N-i+1} = \mathrm{mod}(IG_{M\times N-i+1} + CG_{M\times N-i+2}, 256) \oplus K_i^5 \oplus \mathrm{mod}(CR_{i-1} + CB_{i-1}, 256), \quad (24)$$

$$CB_i = \mathrm{mod}(IB_i + CB_{i-1}, 256) \oplus K_i^6 \oplus \mathrm{mod}(CR_{i-1} + CG_{M\times N-i+1}, 256), \quad (25)$$

where CR_0, $CG_{M\times N+1}$, CB_0 are initial values. In the diffusion process, we can get two totally different encrypted images even though we change the last pixel because IG is diffused in reverse order.

Step 8. Combine the CR, CG and CB components to obtain cipher-image C.

Figure 2 shows the encryption result of the plain-image whose size is 480×320.

2.4. *Proposed decryption algorithm*

The decryption algorithm is the inverse process of the encryption algorithm. The receiver party obtains chaotic sequences by having appropriate initial conditions x_1, x_2, x_3, x_4, x_5 that the sender sends. And in the paper, different images will produce different initial conditions because the initial conditions are produced by plain-images. After that, we can get three matrices CR, CG and CB by dividing the color cipher-image into red, green and blue components. The detailed decryption scheme is presented by the follow steps:

Step 1. Iterate the hyperchaotic system (1) using keys that the sender sends to gain the chaotic sequences S^1, S^2, S^3, S^4 and S^5 according to Sec. 2.2.

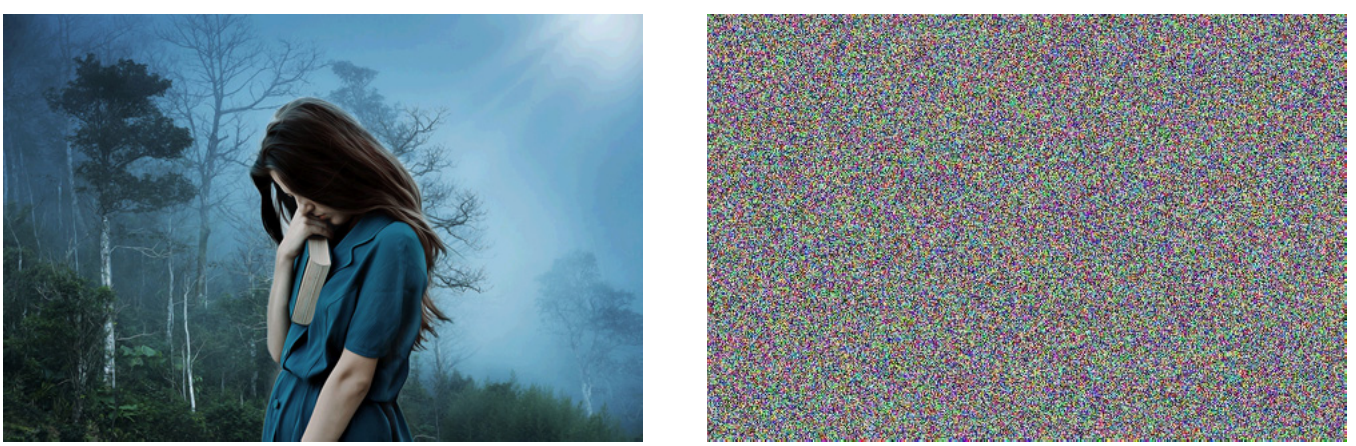
Step 2. Modify the original initial values by the formulae (15)–(19). And iterate the hyperchaotic system (1). Then, we can get three chaotic sequences with length MN: $K^1 = \{k_1^1, k_2^1, \ldots, k_{M\times N}^1\}$, $K^2 = \{k_1^2, k_2^2, \ldots, k_{M\times N}^2\}$, $K^3 = \{k_1^3, k_2^3, \ldots, k_{M\times N}^3\}$.

Step 3. Obtain the key streams $K^4 = \{k_1^4, k_2^4, \ldots, k_{M\times N}^4\}$, $K^5 = \{k_1^5, k_2^5, \ldots, k_{M\times N}^5\}$, $K^6 = \{k_1^6, k_2^6, \ldots, k_{M\times N}^6\}$ by processing the chaotic sequences K^1, K^2, K^3 according to the formulae (20)–(22).

Step 4. We can obtain the intermediate images by executing the inverse process of diffusion according to the following formulae (26)–(28).

$$IR_i = \mathrm{mod}((CR_i \oplus K_i^4 \oplus \mathrm{mod}(CG_{M\times N-i+1} + CB_{i-1}, 256)) + 256 - CR_{i-1}, 256), \quad (26)$$

$$IG_i = \mathrm{mod}((CG_{M\times N-i+1} \oplus K_i^5 \oplus \mathrm{mod}(CR_{i-1} + CB_{i-1}, 256)) + 256 - CG_{M\times N-i+2}, 256), \quad (27)$$



(a) (b)

Fig. 2. (a) The plain-image (480×320) and (b) the cipher-image.

$$IB_i = \mathrm{mod}((CB_i \oplus K_i^6 \oplus \mathrm{mod}(CR_{i-1} + CG_{M\times N-i+1}, 256)) + 256 - CB_{i-1}, 256). \quad (28)$$

Step 5. The IR, IG and IB images are divided into nonoverlapping blocks of equal size, i.e. M blocks which have N pixels.

Step 6. Exchange pixels of IG with IB according to the chaotic sequences S^4 and S^5 by the following formula (29).

$$\begin{cases} \mathrm{temp} = IB\{i\}(j), \\ IB\{i\}(j) = IG\{S^4(j)\}(S^5(i)), \\ IG\{S^4(j)\}(S^5(i)) = \mathrm{temp}. \end{cases} \quad (29)$$

Step 7. Exchange pixels of IR with IG and IB according to the chaotic sequences S^1, S^2 and S^3 by the following formulae (30) and (31).

$$\begin{cases} G\{S^2(j)\}(S^3(i)) = IR\{i\}(j), \\ R\{i\}(j) = IG\{S^2(j)\}(S^3(i)), \end{cases} \quad \text{if } S^1((i-1)*N+j) > 0.5, \quad (30)$$

$$\begin{cases} R\{i\}(j) = IB\{S^2(j)\}(S^3(i)), \\ B\{S^2(j)\}(S^3(i)) = IR\{i\}(j), \end{cases} \quad \text{if } S^1((i-1)*N+j) \le 0.5. \quad (31)$$

Step 8. We can obtain the R, G and B plain-images by combining the M matrices, respectively. Then, we can get plain-image P by combining the R, G and B components.

3. Performance and Security Analysis

3.1. *Statistical analysis*

3.1.1. *Histogram*

The histogram of an image reveals the distribution information of pixel values. The pixels of cipher-image should distribute evenly and have a completely different histogram with the plain-image to prevent the enemy from getting any meaningful information of the plain-image.

Figure 3 demonstrates that the histograms of cipher-images are almost uniform and totally different from the histograms of the plain-image. Therefore, attackers find it difficult to obtain any useful statistical information to decrypt the cipher-image in the algorithm.

In addition, we employ variance analysis of histogram to evaluate the uniformity of ciphered images [Zhang & Wang, 2014]. The higher the uniformity of ciphered images, the lower the value of variances of histogram. The variance and Standard Deviation (SD) of histograms are calculated by the following formula (32):

$$\begin{cases} \mathrm{var(Z)} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}(z_i - \mathrm{Z_{mean}})^2, \\ \mathrm{SD(Z)} = \sqrt{\mathrm{var(Z)}}, \end{cases} \quad (32)$$

where Z is the vector of the histogram values. z_i is the value of pixels in components R, G, B respectively. SD(Z) is the average fluctuation of histogram. The plaintext images of Lena is tested in experiments. In Table 1, the variance values of histograms of ciphered Lena compared with other encryption algorithms [Zhang & Wang, 2014; Zhu *et al.*, 2011] are listed. Table 1 indicates that variance value of our proposed algorithm is about 5200, so the average fluctuation of value of pixels is about 72. Compared with other encryption schemes, the result of our algorithm is smaller. Therefore, the proposed algorithm is efficient to prevent attackers from obtaining any useful statistical information to decrypt the cipher-image in the algorithm.

3.1.2. *Correlation of adjacent pixels*

A secure image encryption scheme should reduce the correlations of adjacent pixels in three directions. The correlation coefficients are described by formulae (33)–(36):

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)D(y)}}, \quad (33)$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad (34)$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2, \quad (35)$$

$$\mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)). \quad (36)$$

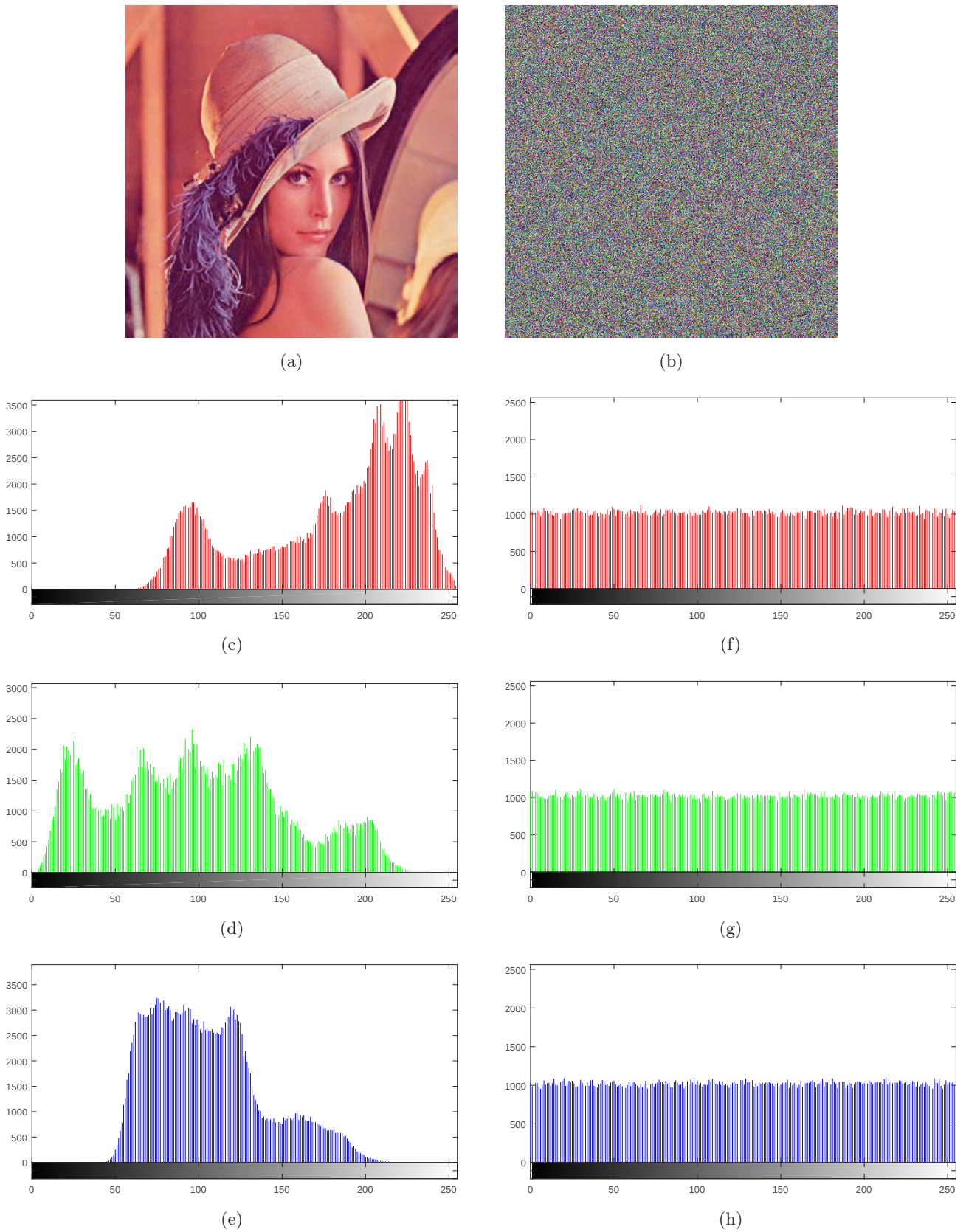


Fig. 3. Histogram analysis: (a) Plain-image of Lena (512×512); (b) cipher-image; (c) red components of the plain-image; (d) green components of the plain-image; (e) blue components of the plain-image; (f) red components of the cipher-image; (g) green components of the cipher-image and (h) blue components of the cipher-image.

Table 1. Variance of histograms.

Image	Encryption Algorithm	Variance
Original image (Lena)		625 571.4908
Encrypted image (Lena)	Ours	5217.9615
	Wang's	5236.5436
	Zhu's	5554.8293

The x and y represent gray-level values of two adjacent pixels. To analyze the correlation of adjacent pixels in the plain-image and corresponding cipher-image, we randomly choose 5000 pairs of adjacent pixels in horizontal, vertical and diagonal directions from the plain-image and corresponding cipher-image, respectively. Then, we can obtain the correlation coefficients in different directions by Eqs. (33)–(36). The consequences of the correlation coefficients are shown in Table 2. The distribution of adjacent pixels of the plain-image and the corresponding cipher-image is given in Fig. 4. It is observed that the plain-image has high correlation coefficients in different directions. However, the correlation coefficients of the cipher-image are very low and the correlation distribution among adjacent pixels of the cipher-image is nearly uniform. Besides, the correlation coefficients among adjacent pixels of the proposed algorithm are compared with those of the other encryption algorithms [Akhshani *et al.*, 2012; Ye *et al.*, 2015; Wang *et al.*, 2016c; Mollaeefar *et al.*, 2015; Wang *et al.*, 2015a; Wang & Luan, 2013] as given in Table 3. It is obvious that our algorithm is better than the other algorithms. Hence, our encryption algorithm possesses better ability of resisting statistical attack compared with other chaos-based color image encryption algorithms.

3.2. *Sensitivity analysis*

3.2.1. *Differential attack*

A good image encryption algorithm should be sensitive to a slight change of the plain-image. It means that we can get a totally different cipher-image even though one bit is changed in plain-image. We usually use two standards to evaluate the effect when one pixel is changed in plain-image. They are the number of pixels change rate (NPCR) and the unified average changing intensity (UACI) which are described respectively as follows:

$$\mathrm{NPCR}_{R,G,B} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} D_{R,G,B}(i,j)}{M \times N} \times 100\%, \quad (37)$$

$$\mathrm{UACI}_{R,G,B} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} |C'_{R,G,B}(i,j) - C_{R,G,B}(i,j)|}{M \times N} \times 100\%, \quad (38)$$

where $C_{R,G,B}$ is the corresponding cipher-image of the plain-image and $C'_{R,G,B}$ is cipher-image of the plain-image when one pixel is changed. $C_{R,G,B}(i,j)$ represents the pixel value at position (i,j) in the corresponding cipher-image, so does $C'_{R,G,B}$.

Table 2. Correlation coefficients of adjacent pixels.

Image	Color Components	Direction		
		Horizontal	Vertical	Diagonal
Original Lena image	Red	0.9783	0.9882	0.9674
	Green	0.9680	0.9812	0.9545
	Blue	0.9335	0.9542	0.9181
Encrypted Lena image	Red	−0.0070	0.0049	−0.0083
	Green	−0.0076	0.0062	0.0032
	Blue	0.0047	−0.0025	−0.0058
Original Peppers image	Red	0.9108	0.9370	0.8987
	Green	0.9320	0.9156	0.9071
	Blue	0.9376	0.8945	0.9029
Encrypted Peppers image	Red	0.0002	0.0131	−0.0015
	Green	−0.0032	0.0072	0.0071
	Blue	−0.0120	0.0080	0.0103

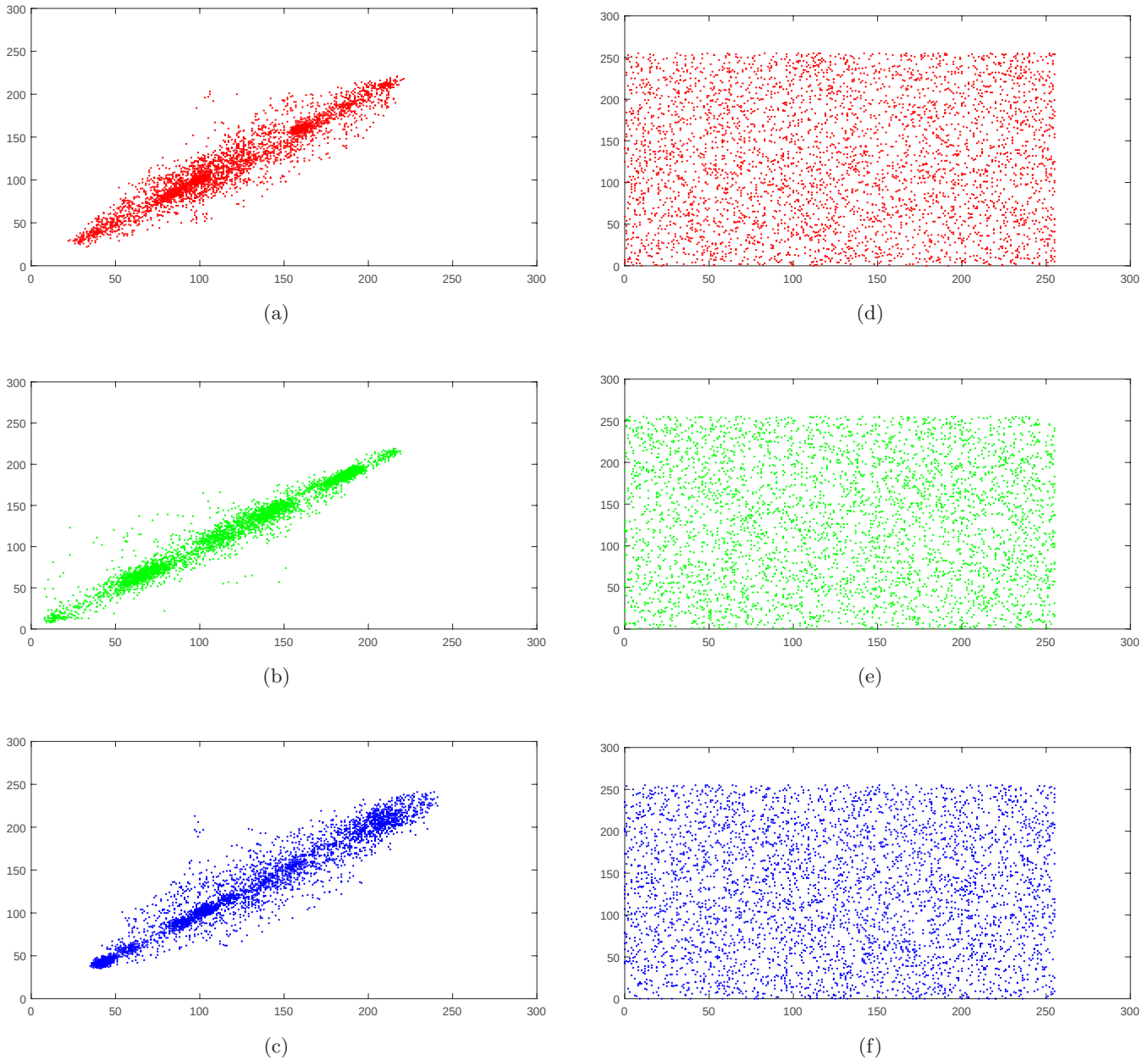


Fig. 4. Correlation distributions of Lena: (a) Red component in horizontal plain-image; (b) green component in vertical plain-image; (c) blue component in the diagonal of plain-image; (d) red component in horizontal cipher-image; (e) green component in vertical cipher-image and (f) blue component in the diagonal of cipher-image.

$D_{R,G,B}$ is described by formula (39).

$$\begin{cases} D_{R,G,B}(i,j) = 1 & \text{if } C_{R,G,B} \neq C'_{R,G,B}, \\ D_{R,G,B}(i,j) = 0 & \text{if } C_{R,G,B} = C'_{R,G,B}. \end{cases} \tag{39}$$

To analyze the sensitivity of the plain-image, firstly, we encrypt the plain-image to obtain a cipher-image. Secondly, we randomly select one pixel in the plain-image and change it. Thirdly, another cipher-image is produced by encrypting the plain-image which has been changed. Finally, we calculate $\text{NPCR}_{R,G,B}$ and $\text{UACI}_{R,G,B}$ by utilizing two encrypted images according to Eqs. (37) and (38). Five hundred values are obtained by calculating the $\text{NPCR}_{R,G,B}$ and $\text{UACI}_{R,G,B}$ with different images, respectively. The values of $\text{NPCR}_{R,G,B}$ and $\text{UACI}_{R,G,B}$ are given in Table 4 which shows that the average values of NPCR and UACI are very close to the ideal values. It means that our encryption algorithm is very susceptive to

Table 3. Correlation coefficients among adjacent pixels in our algorithm compared with some other algorithms.

Image	Encryption Algorithm	Direction		
		Horizontal	Vertical	Diagonal
Original image (Lena)		0.9567	0.9611	0.9169
Encrypted image (Lena)	Ours	0.0064	0.0045	0.0057
	Akhshani's	0.0043	0.0055	0.0072
	Ye's	−0.0175	−0.0175	−0.0183
	Wang's	0.0010	0.0181	0.0061
	Mollaeefar's	−0.0065	0.0006	0.0054
	Gu's	0.0096	0.0342	0.0205
	Luan's	0.0011	0.0193	0.0045

Table 4. Average NPCR and UACI of different color images.

Image	Average NPCR			Average UACI		
	Red	Green	Blue	Red	Green	Blue
Lena	99.6404	99.6334	99.6470	33.4885	33.4930	33.5089
Peppers	99.6615	99.6420	99.6389	33.5691	33.4534	33.4964
Baboon	99.6325	99.6367	99.6438	33.4568	33.4761	33.6113
Airplane	99.6575	99.6341	99.6374	33.5482	33.5035	33.4518

the plain-image and can withstand the differential attacks. By comparing the NPCR and UACI in our algorithm with those in other algorithms [Ye *et al.*, 2015; Belazi *et al.*, 2016b; Wang *et al.*, 2016c; Mollaeefar *et al.*, 2015; Wang *et al.*, 2015c], we can see that our algorithm is better than the other algorithms. Hence, our encryption algorithm possesses a better ability of withstanding differential attacks compared with other chaos-based color image encryption schemes. The consequences are given in Table 5.

3.2.2. *Key sensitivity*

Key sensitivity is an important feature for a security image encryption algorithm which guarantees the security of the algorithm against brute-force attack. A slight change of the plain-image will produce two totally different encrypted images. And a slight change of the secret key should generate two completely different encrypted images as well. The cipher-image cannot be decrypted correctly by modified secret key which has a slight change.

Figure 5 shows the sensitivity of the algorithm we propose to the secret key. Figure 5(c) is the decrypted image with the correct secret key. Figure 5(d) is the decrypted image with the modified secret key. We can see that we cannot obtain the correct plain-image even though the secret key is changed in the last digit after the decimal point of the first parameter. In addition, when changing an initial condition, we do the NPCR and UACI tests to quantify key sensitivity in Table 6. Hence, the hyperchaotic image encryption algorithm we propose is sensitive enough to the secret key.

3.3. *Chosen-plaintext analysis and known-plaintext analysis*

In our algorithm, we will get totally different chaotic sequences and key streams when the plain-image is changed because the initial conditions of the hyperchaotic system are generated by plain-image. The attackers cannot decrypt a cipher-image by the key streams decoded from other images. Hence,

Table 5. Average $\mathrm{NPCR}_{R,G,B}$ and $\mathrm{UACI}_{R,G,B}$ of Lena image in our algorithm compared with some other algorithms.

Algorithm	Average $\mathrm{NPCR}_{R,G,B}$ (%)	Average $\mathrm{UACI}_{R,G,B}$ (%)
Ours	99.6403	33.4968
Ye's	99.6244	33.6994
Belazi's	99.6448	33.5319
Wang's	99.6358	33.4428
Mollaeefar's	99.6126	33.4648
Wang's	99.5864	33.2533

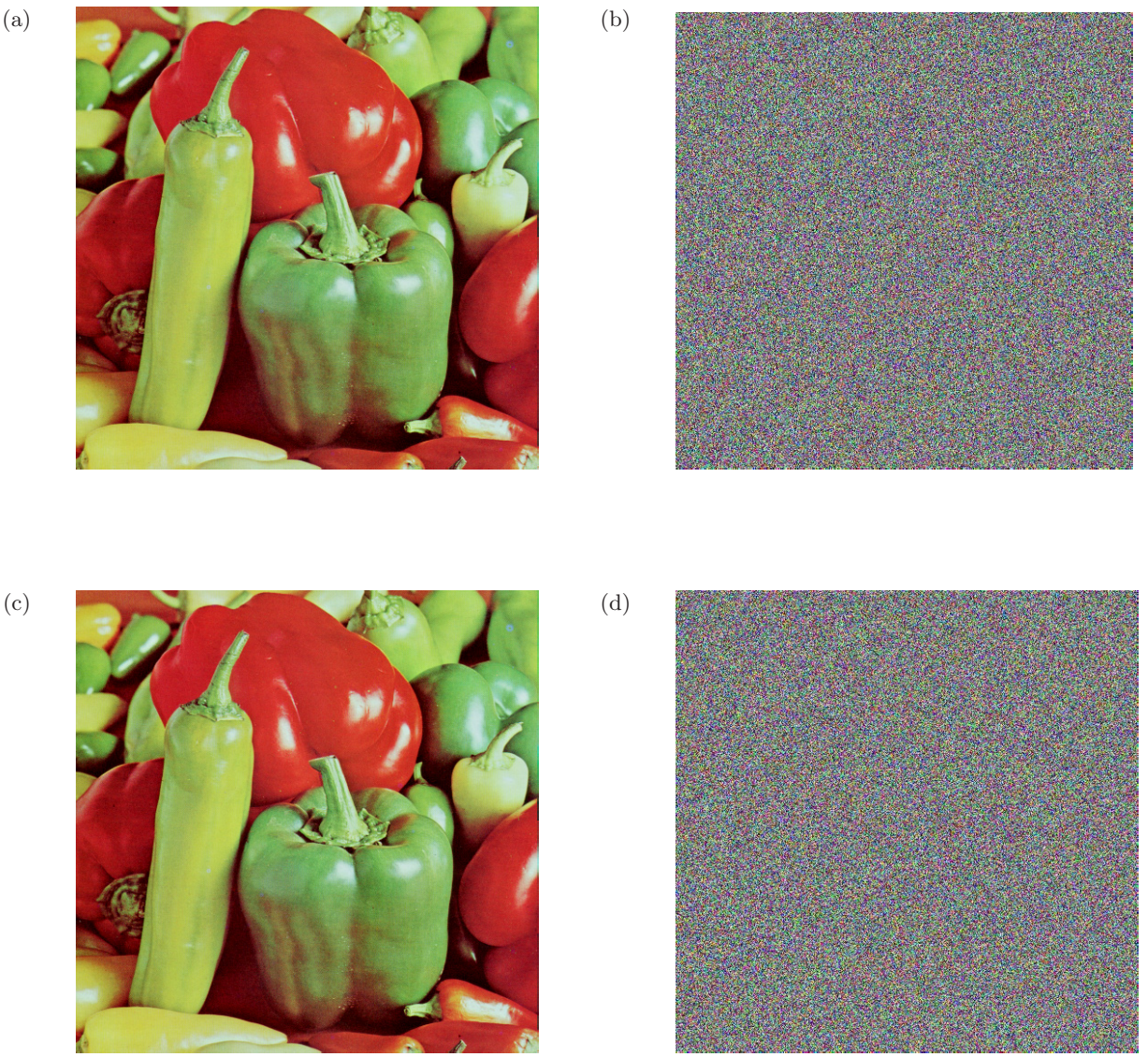


Fig. 5. (a) Plain-image; (b) cipher-image; (c) decrypted image with the correct secret key and (d) decrypted image with modified secret key.

Table 6. Quantified Key Sensitivity: NPCR and UACI tests.

Image		x_1	x_2	x_3	x_4	x_5	Average
Lena (1024 × 1024)	NPCR	99.6141	99.5973	99.6065	99.6317	99.6129	99.6125
	UACI	33.4862	33.5024	33.4372	33.3985	33.4438	33.4536
Lena (512 × 512)	NPCR	99.5946	99.6125	99.6071	99.6014	99.6328	99.6097
	UACI	33.5107	33.5073	33.4772	33.4486	33.4963	33.4880
Lena (256 × 256)	NPCR	99.5833	99.6125	99.6022	99.5927	99.6274	99.6036
	UACI	33.4620	33.4936	33.5104	33.4150	33.5031	33.4768
Pepper (512 × 512)	NPCR	99.6047	99.5821	99.5906	99.6104	99.6217	99.6019
	UACI	33.5135	33.4938	33.4687	33.5022	33.4726	33.4902

Table 7. Key space in our algorithm compared with some other algorithms.

Algorithm	Ours	Gao's	Amin's	Seyedzadeh's
Key space	2^{260}	2^{186}	2^{256}	2^{128}

our encryption scheme can withstand both chosen-plaintext and known-plaintext attacks.

3.4. *Key space analysis*

A large key space is an important feature for a security image encryption algorithm. Key space is composed of initial conditions of the hyperchaotic system and seed keys. The size of the key space should be larger than 2^{100} to provide a high level of security. For the most used double-precision floating-point format which usually refers to binary64, the key space (as specified by the IEEE 754 standard) is 2^{52}. And five initial conditions and three seed keys are considered, the total key space of the proposed scheme is about 2^{260}. The key space of the proposed algorithm is compared with those of other image encryption algorithms [Gao & Chen, 2008; Amin, 2015; Seyedzadeh *et al.*, 2015] in Table 7. The table shows that the key space of the proposed algorithm is enough to withstand any brute-force attack.

3.5. *Information entropy*

Information entropy is an important characteristic to evaluate a good encryption scheme. If the information entropy is more close to 8, the pixels of the image will be distributed more evenly. And the cryptosystem will reveal information as little as possible. The entropy $H(s)$ is calculated by formula (40).

$$H(s) = -\sum_{i=0}^{2^M-1} P(s_i)\log_2 P(s_i), \tag{40}$$

where $P(s_i)$ represents the probability of s_i and 2^M is the number of the information source. To analyze the information entropy of the proposed algorithm, Table 8 provides the results of information entropy. We see that the information entropy of our algorithm is closer to 8 compared with other similar algorithms [Tong *et al.*, 2016b; Amin, 2015; Mollaeefar *et al.*, 2015; Wang *et al.*, 2015c].

3.6. *Complexity analysis*

In our algorithm, the computational cost consists of the permutation process and the diffusion process. In permutation process, a chaotic sequence with length $M \times N$ is produced by the chaotic system. Hence, the computational cost is $\Theta(M \times N)$. Then, we exchange pixels of R with G and B. Hence, the time cost is $\Theta(M \times N)$. Besides, we exchange pixels of G and B. The time cost is also $\Theta(M \times N)$. Therefore, the total time complexity of the permutation process is $\Theta(3 \times M \times N)$. In diffusion process, we obtain three chaotic sequences with $M \times N$ by the chaotic system. Hence, the computational cost is $\Theta(3 \times M \times N)$. And we change pixels of R, G and B respectively. The total time cost is $\Theta(3 \times M \times N)$. Hence, the total time cost of the diffusion process is $\Theta(6 \times M \times N)$. Therefore, the total time complexity is $\Theta(9 \times M \times N)$.

In Table 9, we see the results of the time complexity. Compared with Zhang's algorithm [Zhang & Wang, 2015] and Wang's algorithm [Wang *et al.*, 2016b], the time complexity of the proposed

Table 8. Information entropy in our algorithm compared with some other algorithms.

		Plain-Image			Cipher-Image		
Algorithm	Image	Red	Green	Blue	Red	Green	Blue
Ours	Lena	7.2531	7.5940	6.9684	7.9991	7.9993	7.9993
	Peppers	7.3388	7.4963	7.0583	7.9994	7.9994	7.9993
	Baboon	7.7067	7.4744	7.7522	7.9993	7.9994	7.9992
	Airplane	6.7178	6.7990	6.2138	7.9993	7.9994	7.9993
Tong's	Lena				7.9972	7.9972	7.9976
Amin's	Lena				7.9972	7.9973	7.9971
Mollaeefar's	Lena				7.9914	7.9907	7.9907
Wang's	Lena				7.9971	7.9975	7.9972

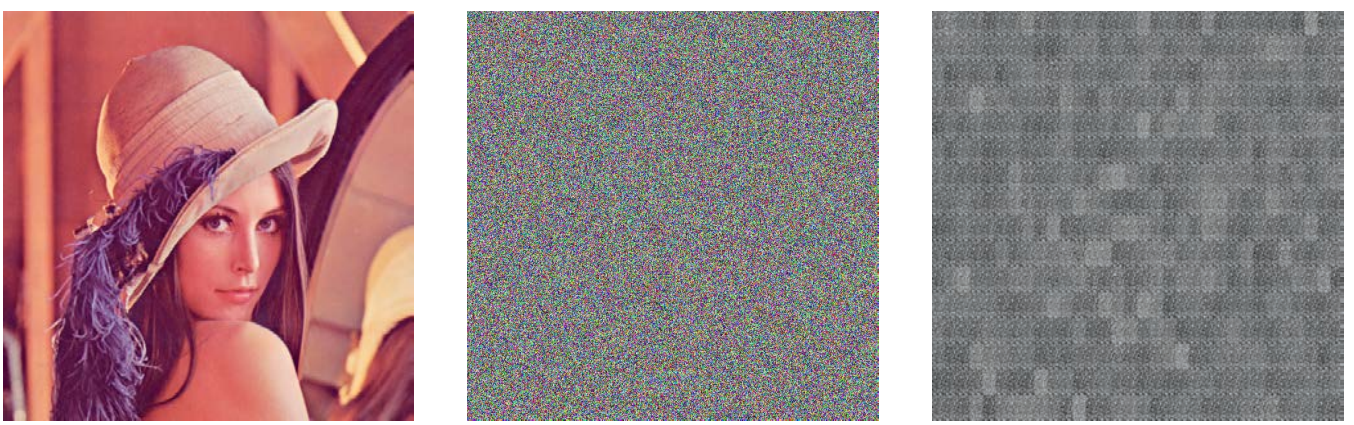
Table 9. Time complexity.

Algorithm	Ours	Zhang's	Wang's
Key space	$\Theta(9 \times M \times N)$	$\Theta(100 \times M \times N)$	$\Theta(12 \times M \times N)$

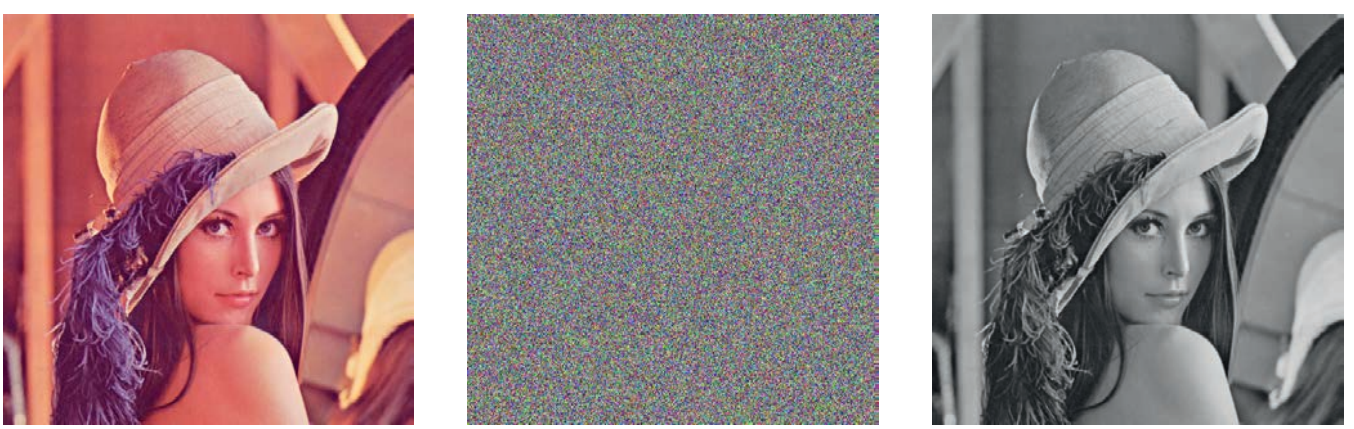
algorithm is lower. Hence the speed of our encryption scheme is faster.

3.7. *Attack test*

For the former color image encryption schemes proposed, if the grayscale image is cracked successfully by the adversary, then the color image will also be cracked soon. But it turns out to be a failure that the adversary used the gray image to crack our algorithm, because the proposed scheme can mix R, G and B components to strengthen the dependence of each component, which is more efficient than former algorithms. Figures 6(a) and 6(d) are the original images of Lena, Figs. 6(b) and 6(e) are the encrypted images of Lena using proposed algorithm and Yin's algorithm [Yin & Wang, 2018], Figs. 6(c) and 6(f) are the decrypted image the enemy gets when the enemy uses the grayscale image to crack our algorithm and Yin's algorithm although they know the key. The experiment shows that the proposed algorithm is more efficient in resisting gray image attack.



(a) (b) (c)



(d) (e) (f)

Fig. 6. Attack test of Lena: (a) Plain-image of Lena (512×512); (b) encrypted image of Lena using proposed algorithm; (c) decrypted image the enemy gets; (d) plain-image of Lena (512×512); (e) encrypted image of Lena using Yin's algorithm and (f) decrypted image the enemy gets.

4. Conclusion

Digital image is one of the most popular forms of multimedia, and it is widely used in politics, economy, national defense, education and so on. In some specific areas such as military, business and medical treatment, digital images also have high confidential requirements. Therefore, some scholars have proposed a number of image encryption schemes. However, in these encryption schemes, the security is not good due to their inappropriate cryptographic structure. In the paper, a novel color image encryption algorithm using hyperchaotic system and block permutation is proposed. A block permutation which is realized by mixing R, G and B components is employed to strengthen the dependence of each component. Besides, it can reduce time consumption. Then, we diffuse the pixels of color components and the three components affect each other again. And in the diffusion process, we diffuse the G component in reverse order. Therefore, we can get two totally different encrypted images even though we change the last pixel. The experimental results reveal that our algorithm possesses better abilities of resisting statistical attacks and differential attacks, with larger key space, closer information entropy to 8, faster encryption speed compared with other chaos-based color image encryption algorithms.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. 61571185 and 61672216), the Science and Technology Planning Project of Hunan Province (2017GK4009), and the Open Fund Project of Key Laboratory in Hunan Universities (No. 18K010).

References

Akhshani, A., Akhavan, A., Lim, S. C. & Hassan, Z. [2012] "An image encryption scheme based on quantum logistic map," *Commun. Nonlin. Sci. Numer. Simul.* **17**, 4653–4661.

Amin, Z. [2015] "Complex dynamics in a 5D hyperchaotic attractor with four-wing, one equilibrium and multiple chaotic attractors," *Nonlin. Dyn.* **81**, 585–605.

Bakhshandeh, A. & Eslami, Z. [2013] "An authenticated image encryption scheme based on chaotic maps and memory cellular automata," *Opt. Laser. Engin.* **51**, 665–673.

Baptista, M. S. [1998] "Cryptography with chaos," *Phys. Lett. A* **250**, 50–54.

Belazi, A., El-Latif, A. A. A. & Belghith, S. [2016a] "A novel image encryption scheme based on substitution-permutation network and chaos," *Sign. Process.* **128**, 155–170.

Belazi, A., Khan, M., El-Latif, A. A. A. & Belghith, S. [2016b] "Efficient cryptosystem approaches: S-boxes and permutation-substitution-based encryption," *Nonlin. Dyn.* **87**, 1–25.

Chen, W., Quan, C. & Tay, C. J. [2009] "Optical color image encryption based on Arnold transform and interference method," *Opt. Commun.* **282**, 3680–3685.

Gao, T. G. & Chen, Z. Q. [2008] "A new image encryption algorithm based on hyperchaos," *Phys. Lett. A* **372**, 394–400.

Hu, T., Liu, Y., Gong, L. H., Guo, S. F. & Yuan, H. M. [2017] "Chaotic image cryptosystem using DNA deletion and DNA insertion," *Sign. Process.* **134**, 234–243.

Hua, Z. Y. & Zhou, Y. C. [2017] "Design of image cipher using block-based scrambling and image filtering," *Inform. Sci.* **396**, 97–113.

Huang, X. & Ye, G. [2014] "An efficient self-adaptive model for chaotic image encryption algorithm," *Commun. Nonlin. Sci. Numer. Simul.* **19**, 4094–4104.

Jakimoski, G. & Kocarev, L. [2001] "Chaos and cryptography: Block encryption ciphers based on chaotic maps," *IEEE Trans. Circuits Syst.-I: Fund. Th. Appl.* **48**, 163–169.

Jawad, L. M. & Sulong, G. [2015] "Chaotic map-embedded blowfish algorithm for security enhancement of colour image encryption," *Nonlin. Dyn.* **81**, 2079–2093.

Li, C. Q., Lin, D. D. & Lu, J. [2017a] "Cryptanalyzing an image-scrambling encryption algorithm of pixel bits," *IEEE MultiMed.* **24**, 64–71.

Li, Y. P., Wang, C. H. & Chen, H. [2017b] "A hyperchaos-based image encryption algorithm using pixel-level permutation and bit-level permutation," *Opt. Laser. Engin.* **90**, 238–246.

Liu, H. & Wang, X. [2010] "Color image encryption based on one-time keys and robust chaotic maps," *Comput. Math. Appl.* **59**, 3320–3327.

Liu, H. & Wang, X. [2011] "Color image encryption using spatial bit-level permutation and high-dimension chaotic system," *Opt. Commun.* **284**, 3895–3903.

Liu, H., Wang, X. & Kadir, A. [2012] "Image encryption using DNA complementary rule and chaotic maps," *Appl. Soft Comput.* **12**, 1457–1466.

Liu, W. H., Sun, K. H. & Zhu, C. X. [2016] "A fast image encryption algorithm based on chaotic map," *Opt. Laser. Engin.* **84**, 26–36.

Mao, Y. B., Chen, G. R. & Lian, S. G. [2004] "A novel fast image encryption scheme based on 3d chaotic baker maps," *Int. J. Bifurcation and Chaos* **14**, 3613–3624.

Mollaeefar, M., Sharif, A. & Nazari, M. [2015] "A novel encryption scheme for colored image based on high level chaotic maps," *Multimed. Tools Appl.* **76**, 1–23.

Norouzi, B. & Mirzakuchaki, S. [2014] "A fast color image encryption algorithm based on hyperchaotic systems," *Nonlin. Dyn.* **78**, 995–1015.

Özkaynak, F. [2018] "Brief review on application of nonlinear dynamics in image encryption," *Nonlin. Dyn.* **92**, 305–313.

Pak, C. & Huang, L. L. [2017] "A new color image encryption using combination of the 1D chaotic map," *Sign. Process.* **138**, 129–137.

Schmitz, R. [2001] "Use of chaotic dynamical systems in cryptography," *J. Franklin Instit.* **338**, 429–441.

Seyedzadeh, S. M., Norouz, B., Mosavi, M. R. & Mirzakuchaki, S. [2015] "A novel color image encryption algorithm based on spatial permutation and quantum chaotic map," *Nonlin. Dyn.* **81**, 1–19.

Som, S., Dutta, S., Singha, R., Kotal, A. & Palit, S. [2015] "Confusion and diffusion of color images with multiple chaotic maps and chaos-based pseudorandom binary number generator," *Nonlin. Dyn.* **80**, 615–627.

Sui, L. S. & Gao, B. [2013] "Color image encryption based on gyrator transform and Arnold transform," *Opt. Laser Technol.* **48**, 530–538.

Tong, X. J. & Cui, M. G. [2009] "Image encryption scheme based on 3d baker with dynamical compound chaotic sequence cipher generator," *Sign. Process.* **89**, 480–491.

Tong, X. J., Chen, P. H. & Zhang, M. [2016a] "A joint image lossless compression and encryption method based on chaotic map," *Multimed. Tools Appl.* **76**, 13995–14020.

Tong, X. J., Zhang, M., Wang, Z. & Ma, J. [2016b] "A joint color image encryption and compression scheme based on hyperchaotic system," *Nonlin. Dyn.* **84**, 2333–2356.

Wang, X., Yang, L., Liu, R. & Kadir, A. [2010] "A chaotic image encryption algorithm based on perceptron model," *Nonlin. Dyn.* **62**, 615–621.

Wang, X. Y., Teng, L. & Qin, X. [2012] "A novel colour image encryption algorithm based on chaos," *Sign. Process.* **92**, 1101–1108.

Wang, X. & Luan, D. [2013] "A novel image encryption algorithm using chaos and reversible cellular automata," *Commun. Nonlin. Sci. Numer. Simul.* **18**, 3075–3085.

Wang, X. Y., Gu, S. X. & Zhang, Y. Q. [2015a] "Novel image encryption algorithm based on cycle shift and chaotic system," *Opt. Laser. Engin.* **68**, 126–134.

Wang, X. Y., Zhang, Y. Q. & Bao, X. M. [2015b] "A novel chaotic image encryption scheme using DNA sequence operations," *Opt. Laser. Engin.* **7**, 53–61.

Wang, X., Liu, L. & Zhang, Y. [2015c] "A novel chaotic block image encryption algorithm based on dynamic random growth technique," *Opt. Laser. Engin.* **66**, 10–18.

Wang, L. Y., Song, H. J. & Liu, P. [2016a] "A novel hybrid color image encryption algorithm using two complex chaotic systems," *Opt. Laser. Engin.* **77**, 118–125.

Wang, X. Y., Zhang, H. L. & Bao, X. M. [2016b] "Color image encryption scheme using CML and DNA sequence operations," *BioSystems* **144**, 18–26.

Wang, X. Y., Zhao, Y. Y., Zhang, H. L. & Guo, K. [2016c] "A novel color image encryption scheme using alternate chaotic mapping structure," *Opt. Laser. Engin.* **82**, 79–86.

Xie, E. Y., Li, C. Q., Yu, S. M. & Lu, J. [2016] "On the cryptanalysis of Fridrich's chaotic image encryption scheme," *Sign. Process.* **132**, 150–154.

Xu, L., Gou, X., Li, Z. & Li, J. [2017] "A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion," *Opt. Laser. Engin.* **91**, 41–52.

Ye, G. D., Zhang, H. Q. & Chai, H. J. [2015] "Chaotic image encryption algorithm using wave-line permutation and block diffusion," *Nonlin. Dyn.* **83**, 1–11.

Yin, Q. & Wang, C. [2018] "A new chaotic image encryption scheme using breadth-first search and dynamic diffusion," *Int. J. Bifurcation and Chaos* **28**, 1850047-1–13.

Yuan, H. M., Liu, Y., Gong, L. H. & Wang, J. [2016] "A new image cryptosystem based on 2D hyperchaotic system," *Multimed. Tools Appl.* **76**, 8087–8108.

Zhang, Y. Q. & Wang, X. Y. [2014] "A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice," *Inform. Sci.* **273**, 329–351.

Zhang, Y. Q. & Wang, X. Y. [2015] "A new image encryption algorithm based on non-adjacent coupled map lattices," *Appl. Soft Comput.* **26**, 10–20.

Zhang, X. P., Nie, W. G., Ma, Y. L. & Tian, Q. Q. [2016] "Cryptanalysis and improvement of an image encryption algorithm based on hyperchaotic system and dynamic S-box," *Multimed. Tools Appl.* **76**, 15641–15659.

Zhang, X. & Wang, C. H. [2019] "A novel multi-attractor period multi-scroll chaotic integrated circuit based on CMOS wide adjustable CCCII," *IEEE Access* **7**, 16336–16350.

Zhou, L., Wang, C. H., Zhang, X. & Yao, W. [2018a] "Various attractors, coexisting attractors and antimonotonicity in a simple fourth-order memristive twin-T oscillator," *Int. J. Bifurcation and Chaos* **28**, 1850050-1–18.

Zhou, L., Wang, C. H. & Zhou, L. L. [2018b] "A novel no-equilibrium hyperchaotic multiwing system via introducing memristor," *Int. J. Circuit Th. Appl.* **46**, 84–98.

Zhu, Z. L., Zhang, W., Wong, K. W. & Yu, H. [2011] "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Inform. Sci.* **181**, 1171–1186.

Zhu, H. G., Zhang, X. D., Yu, H., Zhao, C. & Zhu, Z. L. [2017] "An image encryption algorithm based on compound homogeneous hyperchaotic system," *Nonlin. Dyn.* **89**, 61–79.