

# A robust and lossless DNA encryption scheme for color images

Xiangjun Wu<sup>1,2,3</sup> · Jürgen Kurths<sup>4,3</sup> · Haibin Kan<sup>2</sup>

Received: 8 December 2016 / Revised: 16 April 2017 / Accepted: 29 May 2017 /

Published online: 16 June 2017

© Springer Science+Business Media New York 2017

**Abstract** In this paper, a new robust and lossless color image encryption algorithm is presented based on DNA sequence operation and one-way coupled-map lattices (OCML). The plain-image is firstly decomposed into three gray-level components and we randomly convert them into three DNA matrices by the DNA encoding rules. Then the XOR operation is performed on the DNA matrices for two times. Next, the shuffled DNA matrices are transformed into three gray images according to the DNA decoding rules. Finally, a diffusion process is further applied to change the image pixel's values by a key stream, and the cipher-image is attained. The key stream generated by OCML is related to the plain-image. Experimental results and security analysis demonstrate that the proposed algorithm has a good encryption effect and can withstand various typical attacks. Furthermore, it is robust against some common image processing operations such as noise adding, cropping, JPEG compression etc.

**Keywords** Color image encryption · Lossless · OCML · DNA computing · Robustness

## 1 Introduction

With the rapid development of information sciences and network technologies, a great amount of digital multimedia contents, such as text, images, audio, video etc., are frequently transmitted over the internet. Meanwhile, the security of the digital information becomes a significant and urgent issue. To date, many methods of steganography and cryptography have been

---

✉ Xiangjun Wu  
wuhsiang@yeah.net

✉ Haibin Kan  
hbkan@fudan.edu.cn

<sup>1</sup> College of Software, Henan University, Kaifeng 475004, China

<sup>2</sup> School of Computer Science, Fudan University, Shanghai 200433, China

<sup>3</sup> Department of Physics, Humboldt University, 12489 Berlin, Germany

<sup>4</sup> Potsdam Institute for Climate Impact Research (PIK), 14473 Potsdam, Germany

proposed for protecting multimedia information, for example, DES, AES, RSA, digital watermarking, authentication, digital signature etc. Especially, along with the prevalence of the digital images in various applications, image encryption has attracted increasing attention and been widely studied. Many conventional encryption schemes, such as DES, AES, RSA etc., are typically devised for text. Unfortunately, they are inappropriate for the digital images due to their inherent features, e.g. bulk data capacity, strong correlation among adjacent pixels, high redundancy and so forth.

Chaotic systems are found to have many cryptographically desirable properties of extreme sensitivity to initial conditions and parameters, high randomness, ergodicity and mixing, which makes them suitable for multimedia encryption [16, 27]. Since Fridrich [6] introduced a symmetric image encryption algorithm using the standard Baker map, the researchers have proposed lots of image cryptosystems based on chaos. Compared with traditional ciphers, chaos-based image encryption methods have shown the excellent performance in aspect of security, speed, complexity and computing power. The typical chaos-based ciphers can be divided into two stages, i.e., confusion and diffusion. In the confusion stage, only the positions of pixels from the plain-image are permuted by chaotic sequences or by some matrix transformations, so the histograms of the scrambled image and the original image are identical. The permutation-only algorithms are not secure and threatened by the statistical analysis [4, 6, 12, 19, 20, 38, 59]. In the diffusion stage, the pixel values of the original image are modified by chaotic sequences so that a minute change for one pixel in the plain-image can cause significant changes in the cipher-image. In contrast to the confusion, the diffusion may yield higher security. In practice, confusion and diffusion are usually combined to design the image cryptosystems with the satisfactory performance [42, 49].

According to the chaotic systems used in the encryption algorithms, the image encryption schemes can also be classified into two major categories. One kind of them is based on one-dimensional (1D) or two-dimensional (2D) chaotic maps [13, 22, 33, 43, 48]. Though these chaotic maps are easy to implement, they have some disadvantages including small key space and weak security [57]. Another kind employs higher dimensional (3D, 4D, ..... ) chaotic/hyperchaotic systems [7, 10, 15, 24, 30, 40, 46, 51, 53, 58], multiple chaotic systems [14, 54] and spatiotemporal chaos [35, 39, 44, 50] to design image encryption algorithms. In [8, 18, 25, 32, 34, 37, 55], some attacks such as chosen-ciphertext attack, known-plaintext attack, chosen-plaintext attack, were recommended to break the chaos-based cryptosystems presented in [14, 22, 35, 53, 58]. So it is essential to improve the image encryption techniques for strengthening the security.

In the past years, DNA computing has become a hot topic, and been widely investigated by the researchers from various fields due to its excellent features such as massive parallelism, huge storage and ultra-low power consumption [1, 11, 56]. In the cryptographic field, many DNA-based encryption methods are proposed and implemented by the modern biological technology [47]. For instance, Clelland et al. [3] proposed a steganography scheme by hiding the message in DNA microdots. In [9], for solving the storage problem of one-time pad, an image encryption algorithm was presented based on the one-time pad cryptography with DNA strands. Shyam et al. [36] introduced a new DNA-based encryption method using the nature DNA sequences and the XOR logical operation. For the aforementioned DNA-based encryption approaches, the biological experiments have to be performed in both encryption and decryption processes. However, such experiments can only be conducted in a well equipped laboratory applying current technology, and larger cost is required. To overcome these drawbacks, Kang [29]

developed a novel encryption scheme by the pseudo DNA technique, where real biological experiments are not required. It has better encryption effect, but can only be applied to encrypt character information. Inspired by this idea, many DNA-based image encryption algorithms are proposed in recent years [5, 22, 23, 46, 51–54]. In [22], a RGB image encryption algorithm was designed based on DNA encoding and chaos map. Liu et al. [25] pointed out that this method is very weak to a chosen plain-text attack, and has two other security defects, i.e., low sensitivity with respect to changes of plaintext and security keys. Wei et al. [46] proposed a new color image encryption algorithm based on DNA sequence addition operation and Chen hyperchaotic system. Zhang et al. [53] presented a new image fusion encryption algorithm based on image fusion and DNA sequence operation and hyperchaotic system. The authors in [55] broke this cryptosystem by the chosen plain-text attack. Based on DNA subsequence operation and chaotic system, a couple images encryption algorithm is presented in [51]. But this scheme is weak against differential attacks. The DNA-based encryption algorithms in [5, 23, 52, 54] mainly focus on gray-scale images. In addition, the current image encryption schemes based on DNA computing and chaos seldom study the robustness against some common image processing operations such as noise adding, cropping, JPEG compression, contrast adjustment, histogram equalization etc. However, these operations often occur in real applications. Therefore, it is significant to develop the robust DNA-based encryption algorithms for color images.

Motivated by the above discussions, in this paper, we develop a robust and lossless color image cryptosystem based on DNA computing and OCML. Firstly, the plain-image is arbitrarily encoded according to the DNA encoding rules. Then we employ the DNA XOR operation to implement image encryption, and decode the DNA matrices by the DNA decoding rules. Finally, combining the key stream generated from both OCML and the plain-image, an image diffusion process is further introduced to enhance the security of the image cryptosystem. The proposed color image encryption scheme can be easily extended to handle gray-scale images. Experimental results and security analysis show that the presented image cryptosystem has both good encryption effect and high security. Moreover, an extensive tolerance analysis of some common image processing operations illustrates that our method is highly robust against the attacks of noise adding, cropping, JPEG compression, contrast adjustment, histogram equalization etc.

The rest of this paper is organized as follows. The DNA computing theory, OCML and Logistic map are described briefly in Section 2. Section 3 introduces the proposed color image encryption algorithm in detail. Section 4 provides the experimental results, detailed security and robustness analysis of the presented encryption algorithm. The conclusions are finally drawn in Section 5.

## 2 Preliminaries

### 2.1 DNA sequence and its operations

A Deoxyribonucleic Acid (DNA) is the hereditary material in all known living organisms and many viruses [2, 26]. The information in DNA is preserved as a code consisting of four chemical bases: adenine (A), cytosine (C), guanine (G), and thymine (T). The sequence of these bases determines the information available for building and maintaining an organism [2].

DNA bases pair up with each other, i.e., ‘A’ always pairs with ‘T’, and ‘C’ always pairs with ‘G’. That is, ‘A’ and ‘T’, and ‘C’ and ‘G’ are complementary pairs, respectively. As we know, the binary system is employed to store and process the information in modern computers, where ‘0’ and ‘1’ are complementary. Therefore, it can be concluded that ‘00’ and ‘11’ are complementary, and ‘01’ and ‘10’ are also complementary. If we use the binary numbers ‘00’, ‘01’, ‘10’ and ‘11’ to represent four bases ‘A’, ‘C’, ‘G’ and ‘T’, respectively, there are 24 kinds of coding schemes. However, only eight kinds of them meet the Watson-Crick complementary rule [45], which are given in Table 1.

In our work, the DNA sequences are considered to encode and decode the digital images. Assume that the four bases ‘A’, ‘C’, ‘G’ and ‘T’ are applied to represent the binary numbers ‘00’, ‘01’, ‘10’ and ‘11’, respectively. Hence each 8-bit pixel value of an image can be encoded into a DNA sequence with length 4. For example, if a pixel value of an image is 75, its corresponding binary sequence can be denoted as ‘01001011’. Using different DNA encoding rules listed in Table 1 will result in diverse DNA sequences, i.e., DNA sequence ‘CAGT’ by Rule 1, DNA sequence ‘GACT’ by Rule 2, DNA sequence ‘ACTG’ by Rule 3, DNA sequence ‘AGTC’ by Rule 4, DNA sequence ‘TCAG’ by Rule 5, DNA sequence ‘TGAC’ by Rule 6, DNA sequence ‘CTGA’ by Rule 7, DNA sequence ‘GTCA’ by Rule 8. On the contrary, the same DNA sequence can be decoded as distinct binary sequences by different DNA decoding rules in Table 1. For instance, the DNA sequence ‘ACTG’ can be decoded as a binary sequence ‘01001011’ by Rule 3. However, for the same DNA sequence, another binary sequence ‘00011110’ can be obtained according to Rule 1.

In recent years, DNA computing has drawn much attention in many areas such as molecule computing, data storage, information security etc., owing to huge potential of parallel computing ability, immense information storage density, and ultra-low energy consumption [1]. The researchers have proposed some biological and algebraic operations for the DNA sequences, such as addition, subtraction and exclusive or (XOR) operations [5, 22, 23, 46, 51–54]. In our proposed scheme, the DNA XOR operation is used. The XOR operation for the DNA sequences is defined based on the traditional XOR operation in the binary. There are also eight types of the DNA XOR rules. Table 2 provides one type of the DNA XOR operation, which is employed to encrypt the images in our algorithm. Note that a base in each row or column is unique, namely, the XOR operation has a single result. In addition, the DNA XOR is a reflexive operation.

### 2.2 OCML

Spatiotemporal chaos is chaotic dynamics in spatially extended system, which has more complex behavior and more abundant characteristics than the low-dimensional chaotic systems [35, 39, 44, 50]. Spatiotemporal chaotic systems are frequently constructed by coupled

**Table 1** The encoding and decoding rules for DNA sequences

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
A	00	00	01	01	10	10	11	11
C	01	10	00	11	00	11	01	10
G	10	01	11	00	11	00	10	01
T	11	11	10	10	01	01	00	00

**Table 2** The XOR operation for DNA sequences

XOR	A	C	G	T
A	G	T	A	C
C	T	G	C	A
G	A	C	G	T
T	C	A	T	G

ordinary differential equations (CODE), partial differential equations (PDE), or coupled map lattices (CML) [17]. A CML is a dynamical system with discrete time, discrete space and continuous state. The one-way coupled map lattice (OCML) [35] can be described by

$$x_{n+1}(i) = (1-\varepsilon)f(x_n(i)) + \frac{\varepsilon}{2}[f(x_n(i+1)) + f(x_n(i-1))]$$
 (1)

where  $n = 0, 1, 2, 3, \dots$  is the time index,  $i = 1, 2, 3, \dots, L$  is the lattice site index,  $L$  is the length of OCML, and  $\varepsilon \in (0, 1)$  is a coupling constant. The function  $f(x)$  denotes the Logistic-Sine system (LSS) [57], i.e.,  $f(x) = [\mu x(1-x) + (1-\mu/4)\sin(\pi x)] \bmod 1$ , where  $\mu \in [0, 4]$ . The periodic boundary condition is  $x_n(0) = x_n(L)$ . When  $L = 200$ ,  $\mu = 2.5$  and  $\varepsilon = 0.3$ , the spatiotemporal chaos of system (1) is displayed in Fig. 1. In the following encryption algorithm, OCML (1) will be employed to generate the key stream together with the original plain-images.

### 2.3 Logistic map

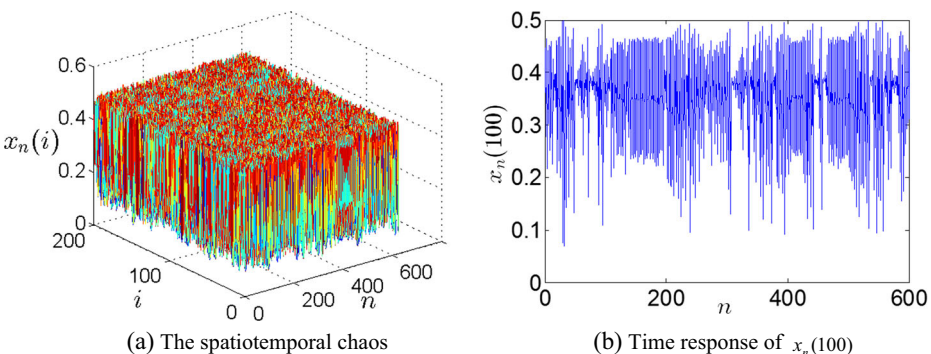
The Logistic map is a well-known 1D map [28], which is defined as follows:

$$y_{n+1} = \delta y_n(1-y_n)$$
 (2)

where  $0 < \delta \leq 4$  and  $y_n \in (0, 1)$ . When  $3.57 \leq \delta \leq 4$ , the Logistic map (2) shows the chaotic behavior. In our work, the Logistic map is utilized to generate some random numbers.

### 3 The proposed algorithm

In this section, the properties and step by step procedures of the proposed method will be described. In the first stage of the presented algorithm, convert the original color



**Fig. 1** The spatiotemporal chaotic dynamics of system (1)

image  $I_0$  into the  $R$ ,  $G$  and  $B$  components and randomly encode the  $R$ ,  $G$  and  $B$  components by the DNA encoding rules in Table 1. We obtain three DNA matrices  $R_1$ ,  $G_1$  and  $B_1$ . In the second stage, we perform the DNA XOR operation on the DNA matrices  $R_1$ ,  $G_1$  and  $B_1$ , and get three DNA matrices  $R_2$ ,  $G_2$  and  $B_2$ . Continue to implement the DNA XOR operation on the DNA matrices  $R_2$ ,  $G_2$  and  $B_2$ , and obtain the DNA matrices  $R_3$ ,  $G_3$  and  $B_3$ . Next, decode the DNA matrices  $R_3$ ,  $G_3$  and  $B_3$ , and transform them into a color image  $I_4$ . Finally, a diffusion process is applied to the image  $I_4$  to enhance the security of the cryptosystem, i.e., the pixel values of the image  $I_4$  are further modified by the key stream  $K$  generated from OCML and the plain-image. Thus we get the encrypted image  $C$ . The flowchart of the proposed encryption algorithm is illustrated in Fig. 2. Without loss of generality, we suppose that the size of the color plain-image  $I_0$  is  $M \times N$ . The proposed algorithm is described in detail as follows.

### 3.1 Generating the random numbers and the key stream

The Logistic map is used to produce the random numbers. Choose arbitrarily an initial condition  $y_0$  and the system parameter  $\delta$ , and then iterate Eq. (2) for  $l + 500$  ( $l \geq 1000$ ) times. Discard the former 500 values and get a chaotic sequence  $Y$  with length  $l$ . The random numbers  $m1$ ,  $m2$  and  $m3$  are generated by the following equations:

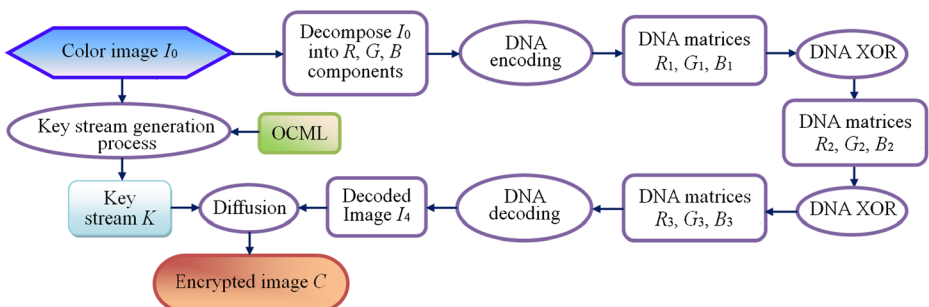
$$m1 = \text{fix}(\text{mod}(Y(t1) \times 10^{t1}, 8) + 1) \tag{3}$$

$$m2 = \text{fix}(\text{mod}(Y(\lfloor l/2 \rfloor) \times 10^{t2}, 8) + 1) \tag{4}$$

$$m3 = \text{fix}(\text{mod}(Y(\lfloor l/3 \rfloor) \times 10^{t3}, 8) + 1) \tag{5}$$

where  $t1, t2, t3 \in [8, 14]$  are positive integers, and  $\text{fix}(y)$  denotes the integer function which rounds the element  $y$  to the nearest integer toward zero.

In the following, we will use the plain-image  $I_0$  and OCML presented in Subsection 2.2 to generate the key stream  $K$ . Corresponding generation procedure is described as follows:



**Fig. 2** The flowchart of the proposed encryption algorithm

(1) Convert the plain-image  $I_0$  into its red, green and blue components, and get three matrices  $R$ ,  $G$  and  $B$  with size  $M \times N$ . Use the following formulae to compute  $\rho_1$ ,  $\rho_2$  and  $\rho_3$ :

$$\rho_1 = \frac{1}{256} \text{mod} \left( \sum_{i=1}^{M/2} \sum_{j=1}^{N/2} B(i, j), 256 \right) \tag{6}$$

$$\rho_2 = \frac{1}{256} \text{mod} \left( \sum_{i=1}^{M/2} R(i, 1) \oplus \sum_{i=1}^M \sum_{j=1}^N G(i, j), 256 \right) \tag{7}$$

$$\rho_3 = \frac{1}{256} \text{mod} \left( \sum_{i=1}^M \sum_{j=1}^N R(i, N) \sum_{i=1}^M \sum_{j=1}^N B(i, j), 256 \right) \tag{8}$$

where the  $\oplus$  operator is the bitwise XOR and  $\&$  denotes the bitwise AND operator.

(2) Choose randomly the initial values  $x_0(\kappa)$  ( $\kappa = 1, 2, 3$ ), and the parameters  $\mu, \varepsilon$  where  $x_0(\kappa) \in (0, 1)$ ,  $\varepsilon \in (0, 1)$  and  $\mu \in [0, 4]$ . The new initial values  $x'_0(\kappa)$  and the new parameters  $\mu', \varepsilon'$  can be obtained by the following equations:

$$x'_0(1) = \text{mod}(x_0(1) + \rho_1, 1) \tag{9}$$

$$x'_0(2) = \text{mod}(x_0(2) + \rho_2, 1) \tag{10}$$

$$x'_0(3) = \text{mod}(x_0(3) + \rho_3, 1) \tag{11}$$

$$\varepsilon' = \text{mod}(\varepsilon + \rho_1 + \rho_3, 1)/10 \tag{12}$$

$$\mu' = \text{mod}(\mu + \rho_1 + \rho_2 + \rho_3, 4) \tag{13}$$

(3) Iterate OCML (1) for  $\tau + MN$  ( $\tau \geq 500$ ) times by using the new initial values  $x'_0(\kappa)$  and the new parameters  $\mu', \varepsilon'$ . To avoid the transient effect, we abandon the former  $\tau$  values. Thus three chaotic sequences with length  $MN$  are obtained, i.e.,  $X_\kappa = \{x_0(\kappa), x_1(\kappa), x_2(\kappa), \dots, x_{MN-1}(\kappa)\}$ .

(4) The sequences  $X_\kappa$  are further improved by the following formula:

$$X'_\kappa(i) = 10^\sigma \times X_\kappa(i) - \text{Int}(10^\sigma \times X_\kappa(i)), \quad (i = 1, 2, \dots, MN) \tag{14}$$

where  $\sigma \in [4, 12]$  is a positive integer, and  $\text{Int}(x)$  returns the element  $x$  to the nearest integer.

(5) Sort the above sequences  $X'_\kappa$  and get three new sequences  $\bar{X}_\kappa$ . Find the position of the values of  $\bar{X}_\kappa$  in  $X'_\kappa$  and mark down the transform positions  $TP_\kappa$ , where  $\bar{X}_\kappa(i)$  is exactly the value of  $X'_\kappa(TP_\kappa(i))$ ,  $\kappa = 1, 2, 3$  and  $i = 1, 2, \dots, MN$ .

(6) Transform  $TP_1$ ,  $TP_2$  and  $TP_3$  into three matrices  $K_1$ ,  $K_2$  and  $K_3$  with size  $M \times N$ , respectively, where  $TP_{\kappa}(i) = K_{\kappa}((\text{mod}(i-1, M) + 1), \lceil i/M \rceil)$ ,  $\kappa = 1, 2, 3$  and  $i = 1, 2, \dots, MN$ . The key stream  $K$  is obtained as follows:

$$\begin{cases} K(1 : M, 1 : N, 1) = K_1(1 : M, 1 : N) \\ K(1 : M, 1 : N, 2) = K_2(1 : M, 1 : N) \\ K(1 : M, 1 : N, 3) = K_3(1 : M, 1 : N) \end{cases} \quad (15)$$

**Remark 1** In our method, the initial conditions  $x_0(1)$ ,  $x_0(2)$ ,  $x_0(3)$ ,  $y_0$ , the system parameters  $\mu$ ,  $\varepsilon$ ,  $\delta$ , and some other parameters  $l$ ,  $t_1$ ,  $t_2$ ,  $t_3$ ,  $\tau$ ,  $\sigma$  are considered as the secret keys.

### 3.2 Image encryption

According to the flowchart shown in Fig. 2, the encryption process is described as follows:

*Step 1.* Input a color plain-image  $I_0$  with size  $M \times N \times 3$ . Decompose the image  $I_0$  into its red, green and blue components, and get three matrices  $R$ ,  $G$  and  $B$  with size  $M \times N$ .

*Step 2.* Generate the key stream  $K$  and three random numbers  $rn1$ ,  $rn2$ ,  $rn3$  through OCML and Logistic map under the initial conditions  $x_0(1)$ ,  $x_0(2)$ ,  $x_0(3)$ ,  $y_0$ , the system parameters  $\mu$ ,  $\varepsilon$ ,  $\delta$ , and the plain-image  $I_0$  based on the method presented in Subsection 3.1.

*Step 3.* Convert the matrices  $R$ ,  $G$  and  $B$  into three binary matrices respectively, then encode randomly the binary matrices by the DNA encoding rules in Table 1 and the random numbers  $rn1$ ,  $rn2$ ,  $rn3$ . We obtain three DNA matrices  $R1$ ,  $G1$  and  $B1$  with size  $M \times 4N$ .

*Step 4.* Perform the XOR operation in Table 2 on the DNA matrices  $R1$ ,  $G1$  and  $B1$  according to the following formulae:

$$\begin{cases} R2(i, j) = R1(i, j) \text{XOR } B1(i, j) \\ G2(i, j) = G1(i, j) \text{XOR } R1(i, j) \\ B2(i, j) = B1(i, j) \text{XOR } R2(i, j) \end{cases} \quad (16)$$

where  $i = 1, 2, \dots, M, j = 1, 2, \dots, 4N$ .

*Step 5.* Carry out the XOR operation in Table 2 on the DNA matrices  $R2$ ,  $G2$  and  $B2$  again by the following formulae:

$$\begin{cases} R3(i, j) = R2(i, j) \text{XOR } G2(i, j) \\ G3(i, j) = G2(i, j) \text{XOR } B2(i, j) \\ B3(i, j) = B2(i, j) \text{XOR } G3(i, j) \end{cases} \quad (17)$$

where  $i = 1, 2, \dots, M, j = 1, 2, \dots, 4N$ .

*Step 6.* Decode the matrices  $R3$ ,  $G3$  and  $B3$  according to the DNA decoding rules in Table 1 and the random numbers  $rn1$ ,  $rn2$ ,  $rn3$ , which transforms the DNA matrices into the binary matrices. Then convert the binary matrices into the decimal matrices  $R4$ ,  $G4$  and  $B4$ , which are separately the red, green and blue components of the image  $I_4$ .



*Step 7.* To enhance the security of the encryption algorithm, we further modify the pixel values of the image  $I_4$  as follows:

$$\begin{cases} C(i, j, 1) = I_4(i, j, 1) \oplus K(i, j, 1) \\ C(i, j, 2) = I_4(i, j, 2) \oplus K(i, j, 2) \\ C(i, j, 3) = I_4(i, j, 3) \oplus K(i, j, 3) \end{cases} \quad (18)$$

where  $i = 1, 2, \dots, M, j = 1, 2, \dots, N$ . The encrypted image  $C$  is finally obtained.

### 3.3 Image decryption

The decryption process is an inverse process of encryption. The steps for the decryption algorithm are given as follows:

*Step 1.* Convert the cipher-image  $C$  into three matrices  $R', G'$  and  $B'$  with size  $M \times N$ .

*Step 2.* Generate the key stream  $K$  and three random numbers  $rn1, rn2, rn3$  through OCML and Logistic map under the initial conditions  $x_0(1), x_0(2), x_0(3), y_0$ , and the system parameters  $\mu, \varepsilon, \delta$ .

*Step 3.* Using the key stream  $K$ , we recover the pixel values of the encrypted image  $C$  by the following formulae:

$$\begin{cases} I1'(i, j, 1) = R1'(i, j) = C(i, j, 1) \oplus K(i, j, 1) \\ I1'(i, j, 2) = G1'(i, j) = C(i, j, 2) \oplus K(i, j, 2) \\ I1'(i, j, 3) = B1'(i, j) = C(i, j, 3) \oplus K(i, j, 3) \end{cases} \quad (19)$$

where  $i = 1, 2, \dots, M, j = 1, 2, \dots, N$ .

*Step 4.* Convert the matrices  $R1', G1'$  and  $B1'$  into three binary matrices respectively, then encode randomly the binary matrices by the DNA encoding rules in Table 1 and the random numbers  $rn1, rn2, rn3$ . We get three DNA matrices  $R2', G2'$  and  $B2'$  with size  $M \times 4N$ .

*Step 5.* Implement the XOR operation in Table 2 on the DNA matrices  $R2', G2'$  and  $B2'$  by the following formulae:

$$\begin{cases} B3'(i, j) = B2'(i, j) \text{XOR } G2'(i, j) \\ G3'(i, j) = G2'(i, j) \text{XOR } B3'(i, j) \\ R3'(i, j) = R2'(i, j) \text{XOR } G3'(i, j) \end{cases} \quad (20)$$

where  $i = 1, 2, \dots, M, j = 1, 2, \dots, 4N$ .

*Step 6.* Perform the XOR operation in Table 2 on the DNA matrices  $R3', G3'$  and  $B3'$  according to the following formulae:

$$\begin{cases} B4'(i, j) = R3'(i, j) \text{XOR } B3'(i, j) \\ R4'(i, j) = B4'(i, j) \text{XOR } R3'(i, j) \\ G4'(i, j) = R4'(i, j) \text{XOR } G3'(i, j) \end{cases} \quad (21)$$

where  $i = 1, 2, \dots, M, j = 1, 2, \dots, 4N$ .

*Step 7.* Transform randomly the DNA matrices  $R4'$ ,  $G4'$  and  $B4'$  into the binary matrices by the DNA decoding rules in Table 1 and the random numbers  $rn1$ ,  $rn2$ ,  $rn3$ , then convert the binary matrices into three decimal matrices  $R5'$ ,  $G5'$  and  $B5'$  with size  $M \times N$ , respectively.  $R5'$ ,  $G5'$  and  $B5'$  are the red, green and blue components of the decrypted image  $I_0'$ .

**Remark 2** The presented encryption scheme in this paper mainly focuses on color images. Further, our method can be easily extended to deal with the gray-scale images. To encrypt the gray-scale images, we perform the following 6 steps:

*Step 1.* Input a original image  $I_0$  with size  $M \times N$ , the initial conditions  $x_0(1)$ ,  $x_0(2)$ ,  $x_0(3)$ ,  $y_0$ , and the system parameters  $\mu$ ,  $\varepsilon$ ,  $\delta$ . Using the generation method given in Subsection 3.1, the Logistic map is used to generate three random numbers  $Rn1$ ,  $Rn2$ ,  $Rn3$ , and OCML is employed to produce a key stream  $K'$  with size  $M \times N$ , where  $K1' = K'(1 : M, 1 : N, 1)$ ,  $K2' = K'(1 : M, 1 : N, 2)$  and  $K3' = K'(1 : M, 1 : N, 3)$ .

*Step 2.* Convert the matrices  $I_0$ ,  $K1'$  and  $K2'$  into the binary matrices  $I_1$ ,  $\Gamma_1$  and  $\Gamma_2$  with size  $M \times 8N$  respectively, then encode randomly the binary matrices  $I_1$ ,  $\Gamma_1$  and  $\Gamma_2$  by the DNA encoding rules in Table 1 and the random numbers  $Rn1$ ,  $Rn2$  and  $Rn3$ . We get three DNA matrices  $I_2$ ,  $A_1$  and  $A_2$  with size  $M \times 4N$ .

*Step 3.* Implement the XOR operation in Table 2 on the DNA matrices  $I_2$  and  $A_2$  based on the following formula:

$$I_3(i, j) = I_2(i, j) \text{XOR } A_2(i, j) \quad (22)$$

where  $i = 1, 2, \dots, M, j = 1, 2, \dots, 4N$ .

*Step 4.* Perform the XOR operation in Table 2 on the DNA matrices  $I_3$  and  $A_1$  by the following formula:

$$I_4(i, j) = I_3(i, j) \text{XOR } A_1(i, j) \quad (23)$$

where  $i = 1, 2, \dots, M, j = 1, 2, \dots, 4N$ .

*Step 5.* Based on the DNA decoding rules in Table 1 and the random numbers  $Rn1$ ,  $Rn2$  and  $Rn3$ , the DNA matrix  $I_4$  is decoded and transformed into a binary matrix  $I_5$ . Then covert the binary matrix  $I_5$  into a decimal matrix  $I_6$ .

*Step 6.* To further enhance the security of the encryption algorithm, we modify the pixel values of the image  $I_6$  by the following formula:

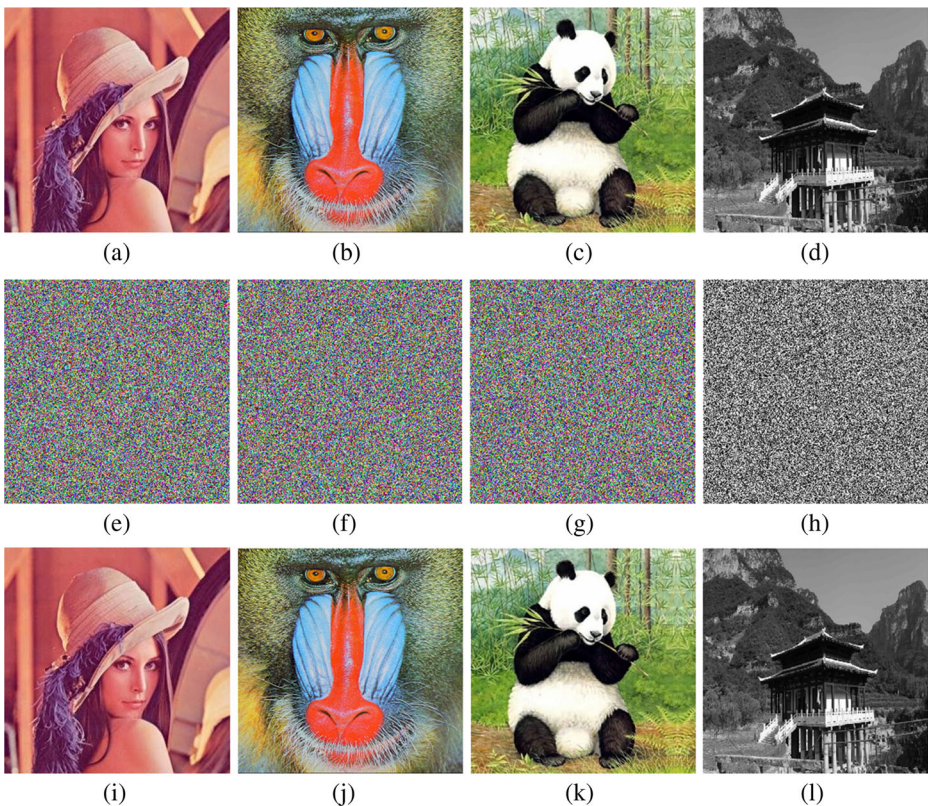
$$C(i, j) = I_6(i, j) \oplus K3'(i, j) \quad (24)$$

where  $i = 1, 2, \dots, M, j = 1, 2, \dots, N$ . Thus the encrypted image  $C$  is obtained.

The decryption procedure of the gray-scale image cryptosystem is similar to that of the above encryption process but just in the reversed order. For saving space, we omit the detailed description of corresponding decryption process.

## 4 Experimental results and performance analysis

In this section, the proposed encryption method is analyzed using different security measures. MATLAB 8.0.0.783 (R2012b) is utilized to simulate the experiment. We use the  $256 \times 256$  color images “Lena”, “Baboon”, “Panda” and the gray-scale image “Tower” shown in Fig. 3(a)–(d) as the original plain-images. Corresponding initial conditions and parameters are chosen arbitrarily as follows:  $x_0(1) = 0.502987342398351$ ,  $x_0(2) = 0.408773287237823$ ,  $x_0(3) = 0.823132423456464$ ,  $y_0 = 0.174373274327838$ ,  $\mu = 3.808387326793071$ ,  $\varepsilon = 0.051234987678346$ ,  $\delta = 3.878237328239921$ ,  $l = 700$ ,  $t1 = 12$ ,  $t2 = 13$ ,  $t3 = 14$ ,  $\tau = 1000$ ,  $\sigma = 6$ . The cipher-images are displayed in Fig. 3(e)–(h), and the decrypted images are shown in Fig. 3(i)–(l). From the results shown in Fig. 3, all cipher-images are noise-like ones and completely unrecognizable. It shows that our proposed algorithm has a good encryption effect. In what follows, the proposed image cryptosystem is analyzed using different security measures and some common image processing operations, i.e., key space analysis, key sensitivity analysis, statistical analysis, Peak Signal-to-Noise Ratio (*PSNR*) analysis, correlation analysis, noise adding, cropping, JPEG compression etc.



**Fig. 3** Experimental results: **a** Plain-image of Lena; **b** Plain-image of Baboon; **c** Plain-image of Panda; **d** Plain-image of Tower; **e** Cipher-image of Lena; **f** Cipher-image of Baboon; **g** Cipher-image of Panda; **h** Cipher-image of Tower; **i** Decrypted image of Lena; **j** Decrypted image of Baboon; **k** Decrypted image of Panda; **l** Decrypted image of Tower

## 4.1 Security key analysis

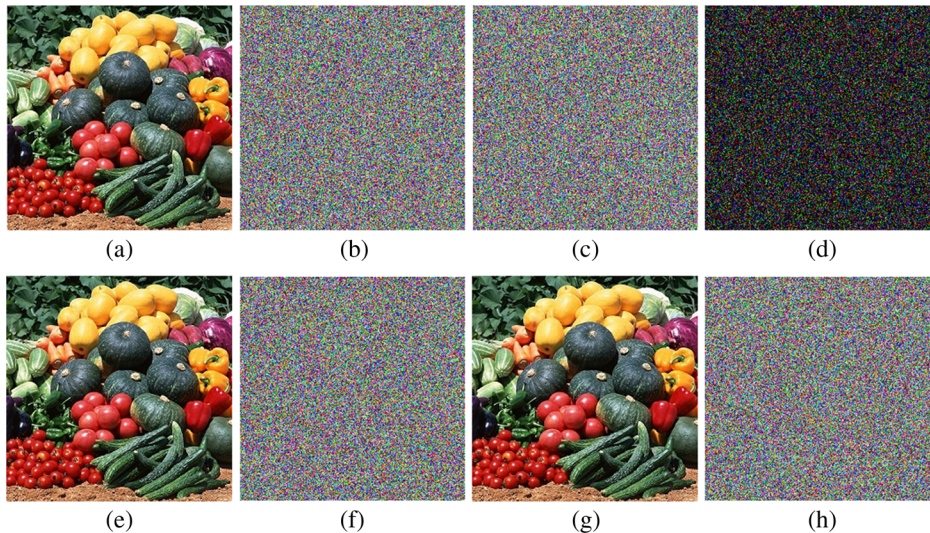
A good image cryptosystem should have a key space large enough to resist the brute-force attacks and should be high sensitive to any change of its secret keys.

### 4.1.1 Key space analysis

As mentioned in Remark 1, the secret keys of the proposed encryption algorithm consist of the initial conditions  $x_0(1), x_0(2), x_0(3), y_0$  and the parameters  $\mu, \varepsilon, \delta, l, t1, t2, t3, \tau, \sigma$ , where  $x_0(1), x_0(2), x_0(3), y_0, \mu, \varepsilon$  and  $\delta$  are all double precision real numbers. If the computational precision is  $10^{-15}$ , the key space of the proposed algorithm is larger than  $10^{105} \approx 2^{349}$ . It is sufficiently large to withstand the brute force attacks.

### 4.1.2 Key sensitive analysis

**Key sensitive test I** We use *KEY0* to denote the initial key set, i.e.,  $KEY0 = \{x_0(1), x_0(2), x_0(3), y_0, \mu, \varepsilon, \delta, l, t1, t2, t3, \tau, \sigma\}$ . *KEY0* is used to encrypt the original image in Fig. 4(a) and corresponding cipher-image is shown in Fig. 4(b). Assume that a tiny change  $\Delta = 10^{-15}$  is applied to the secret keys. For example, let  $\mu' = \mu + \Delta$  while keeping others unchanged, which yields another key set represented by *KEY1*. Figure 4(c) shows the cipher-image by using *KEY1* to encrypt the same original image in Fig. 4(a). The differential image between (b) and (c) is plotted in Fig. 4(d). The difference in terms of pixel gray-scale values between two encrypted images Fig. 4(b) and (c) is 99.62%. So a tiny change in the secret keys will lead to the completely different cipher-images.



**Fig. 4** Key sensitivity test: **a** The original image of Vegetables; **b** The encrypted image of Vegetables with *KEY0*; **c** The encrypted image of Vegetables with *KEY1*; **d** The differential image between (b) and (c); **e** The decrypted image from **b** with the correct key set *KEY0*; **f** The decrypted image from **b** with the wrong key set *KEY1*; **g** The decrypted image from **c** with the correct key set *KEY1*; **h** The decrypted image from **c** with the wrong key set *KEY0*

**Key sensitive test II** The cipher-image cannot be decrypted correctly even though there is a trivial difference between the encryption and decryption keys. The secret key sets  $KEY0$  and  $KEY1$  are separately employed to decrypt the cipher-images in Fig. 4(b) and (c). As shown in Fig. 4(e) and (g), we can see that only when the secret keys in the decryption process are identical to those used in the encryption process, can one recover the original image successfully. However, even a slightly change in the secret keys will result in the failure of image decryption as depicted in Fig. 4(f) and (h).

From the above results, one can find that a minute change of the secret keys will generate a completely different encryption result and cannot recover the plain-image correctly. Therefore, the proposed cryptosystem is highly sensitive to the secret keys.

## 4.2 PSNR analysis

The difference between the original image and the encrypted one is measured by calculating Peak Signal-to-Noise Ratio ( $PSNR$ ) [41].  $PSNR$  is commonly used to evaluate the image quality, which is defined as:

$$PSNR = 20 \log_{10} \left( L / \sqrt{MSE} \right) \quad (25)$$

where

$$MSE = (1/MN) \sum_{i=1}^M \sum_{j=1}^N (I_0(i, j) - I_1(i, j))^2$$

$M$  and  $N$  are the width and the height of the test image, respectively;  $L$  is the maximum possible pixel value of the image;  $I_0(i, j)$  and  $I_1(i, j)$  are the pixel gray values of the original image and the encrypted one, respectively. If the value of  $PSNR$  is bigger than or equal to 30 dB, the human eyes cannot differentiate between the original and processed images. Otherwise, the human visual system is able to perceive the quality degradation. Specially, the  $PSNR$  value is infinite ( $MSE = 0$ ) means that two images  $I_0$  and  $I_1$  are identical.

The values of  $PSNR$  for the test images by using different encryption algorithms are listed in Table 3. As shown in Table 3, using the presented method, the values of  $PSNR$  between the original and the decrypted images are always infinite. It implies that the decrypted image is identical to the original one, namely, our method is a lossless image encryption scheme. The  $PSNR$  values between the original and encrypted images using the proposed scheme are smaller than those using the encryption algorithms reported in [30, 58]. The smaller the  $PSNR$  value, the greater the difference between the original image and the cipher-image, the better the

**Table 3** The  $PSNR$  results between the original images and corresponding ciphered/decrypted images by using different encryption algorithms: ‘O-C’ represents the original and ciphered images, and ‘O-D’ denotes the original and decrypted images

		Lena	Peppers	Baboon	Panda	Vegetables
The proposed algorithm	PSNR (O-D)	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
	PSNR (O-C)	8.1293	7.6393	8.7729	7.7510	7.4482
Ref. [58]	PSNR (O-D)	96.2956	96.2956	96.2956		
	PSNR (O-C)	9.2322	8.9914	9.5466		
Ref. [30]	PSNR (O-C)	8.6878	8.1227	8.8103		

security effect of the encryption algorithm is. Therefore, the proposed approach significantly outperforms those in [30, 58].

### 4.3 Statistical analysis

#### 4.3.1 Histogram analysis

An ideal cryptosystem should be resistance to the statistical attacks. To withstand the histogram analysis attack, it is essential for the cipher-image to have a uniform distribution and no statistical similarity to the original image. Figures 5 and 6 show the histograms of the original image in Fig. 3(a) and corresponding encrypted image in Fig. 3(d), respectively. As can be seen, the histograms of the cipher-image have nearly uniform distribution and are significantly different from the histograms of the plain-image. Thus the presented scheme can resist the histogram analysis attack.

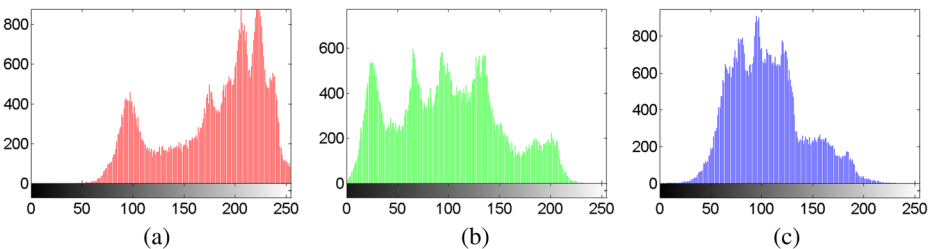
#### 4.3.2 Correlation analysis

As we know, in the visually meaningful images, high correlations exist between pixels and their adjacent pixels either in horizontal, vertical or diagonal direction. An efficient encryption algorithm should break these pixel correlations in the original images, and convert them into noise-like encrypted images with sufficiently low correlations. For an ideal case, there is no correlation between the pixels of the cipher-images. To measure the correlation between two adjacent pixels, we arbitrarily choose 10,000 pairs of adjacent pixels in three directions from the original image and corresponding encrypted image, and calculate the correlation coefficients (*CC*) by the following formulae [4, 59]:

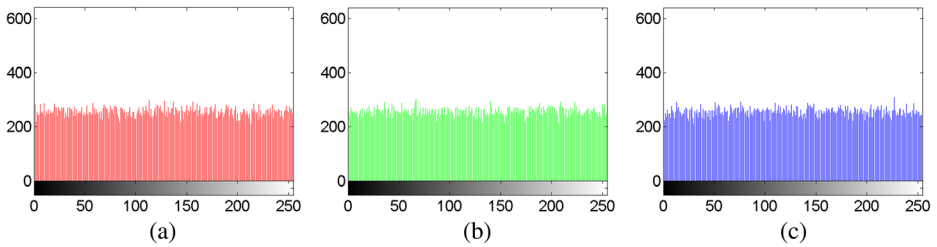
$$CC = \frac{\sum_{i=1}^N \left[ \left( x_i - \frac{1}{N} \sum_{i=1}^N x_i \right) \times \left( y_i - \frac{1}{N} \sum_{i=1}^N y_i \right) \right]}{\sqrt{\left[ \sum_{i=1}^N \left( x_i - \frac{1}{N} \sum_{i=1}^N x_i \right)^2 \right] \times \left[ \sum_{i=1}^N \left( y_i - \frac{1}{N} \sum_{i=1}^N y_i \right)^2 \right]}} \tag{26}$$

where *x* and *y* are grey values of two adjacent pixels in the image.

Figure 7 shows the correlation distribution of adjacent pixels in the plain-image and the encrypted image. One can easily see that the adjacent pixels of the original image cluster together around the diagonal line while the distribution of adjacent pixels of the cipher-image is fairly uniform in the interval [0, 255]. Table 4 shows the results of the correlation coefficients for horizontal, vertical and diagonal adjacent pixels. From Table 4, we can find that the



**Fig. 5** Histograms of the original image of Lena in the **(a)** red, **b** green, **c** blue components

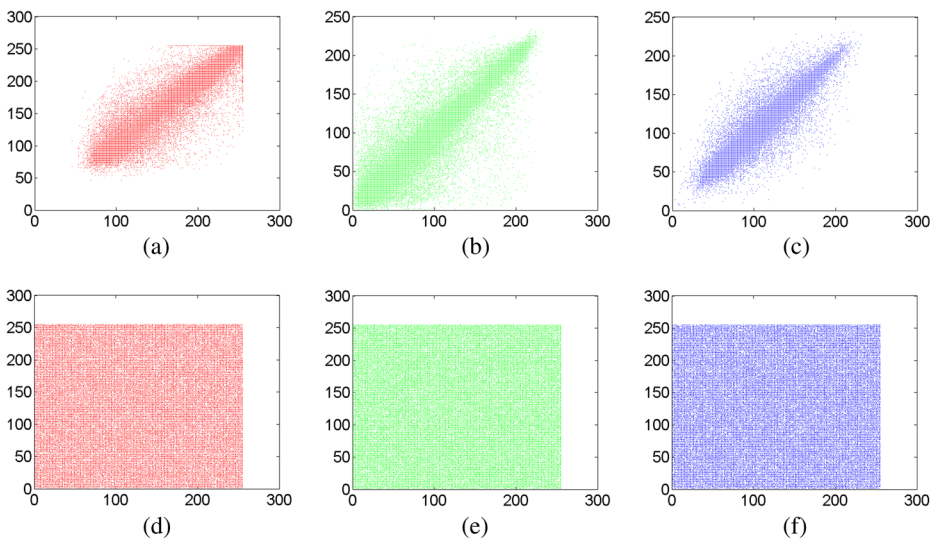


**Fig. 6** Histograms of the cipher-image of Lena in the (a) red, b green, c blue components

correlation coefficients of the original image are close to 1 and the cipher-image has very small correlation values. Obviously, the proposed method has significantly reduced the correlation of adjacent pixels in the encrypted image. Table 5 compares the correlation coefficients of adjacent pixels using the proposed algorithm with those using other encryption schemes in [14, 24, 35, 46, 51]. The results in Table 5 demonstrate that the proposed algorithm has the highest performance in removing the strong correlation between adjacent pixels. So the presented image encryption scheme can efficiently resist the statistical attacks.

#### 4.4 Information entropy analysis

The information entropy is designed to measure the degree of uncertainties in a system [30]. In the image processing applications, the entropy can reflex the distribution of gray-level values in the image. The larger the entropy value of one cipher-image, the more uniform the



**Fig. 7** Correlation distribution of two adjacent pixels: **a** Distribution of vertically adjacent pixels in the red component of the plain Lena image; **b** Distribution of diagonally adjacent pixels in the green component of the plain Lena image; **c** Distribution of horizontally adjacent pixels in the blue component of the plain Lena image; **d** Distribution of vertically adjacent pixels in the red component of the encrypted Lena image; **e** Distribution of diagonally adjacent pixels in the green component of the encrypted Lena image; **f** Distribution of horizontally adjacent pixels in the blue component of the encrypted Lena image

**Table 4** Correlation coefficients of adjacent pixels in the plain-image of Lena and corresponding cipher-image

Lena		Correlation direction		
		Horizontal	Vertical	Diagonal
Original image	Red	0.9244	0.9765	0.9366
	Green	0.9336	0.9794	0.9368
	Blue	0.8545	0.9498	0.9046
Encrypted image	Red	-0.0112	-0.0026	0.0052
	Green	0.0050	0.0199	-0.0064
	Blue	-0.0179	0.0120	-0.0161

distribution of gray-level values in the image, the higher the security is. The information entropy  $H(s)$  of a message source  $s$  is computed as:

$$H(s) = - \sum_{i=0}^{2^N-1} P(s_i) \log_2 P(s_i) \tag{27}$$

where  $P(s_i)$  represents the probability of symbol  $s_i$  and  $2^N$  is the total states of the information source. A truly random source emitting  $2^N$  symbols, the entropy should be  $N$ . Take an ideally random image  $C$  with 256 Gy levels for example, the distribution of pixel intensities in  $[0, 255]$  is uniform, i.e.,  $P(C_i) = 1/256$  where  $i \in [0, 255]$  is a positive integer. Hence the entropy  $H(C) = 8$ .

Table 6 shows the information entropy of the original and encrypted images by using the proposed algorithm and those algorithms in [15, 21, 24, 35]. The averages of the information entropy using the proposed method are also given in the eighth row of Table 6. From these results, the information entropy values of all encrypted images are very close to the theoretical maximum, i.e.,  $H(C) = 8$ . That is, the cipher-image is close to a random source and the probability of divulging information in the encryption process is very little. Furthermore, comparing with other encryption methods [15, 21, 24, 35], the entropy values of the presented cryptosystem are slightly larger. Therefore, the proposed algorithm is more secure against the entropy attack than those algorithms [15, 21, 24, 35].

### 4.5 Differential attack

Generally speaking, a desirable feature of a cipher-image is being sensitive to the slightly changes in a plain-image, for instance, only one pixel of the plain-image is modified. An attacker may try

**Table 5** Comparison of correlation coefficients of adjacent pixels using the proposed algorithm with some other methods

Image	Encryption algorithm	Direction		
		Horizontal	Vertical	Diagonal
Original Lena image		0.9042	0.9686	0.9260
Encrypted Lena image	The proposed algorithm	-0.0080	0.0098	-0.0058
	Ref. [46]	0.0044	0.0034	0.0020
	Ref. [51]	0.0024	0.0012	0.0016
	Ref. [24]	0.0024	0.0580	0.0170
	Ref. [14]	0.1257	0.0581	0.0504
	Ref. [35]	0.0681	0.0845	-



**Table 6** The information entropy of the original and encrypted images by using different encryption algorithms

Encryption algorithms	Test images	Original image			Encrypted image		
		Red	Green	Blue	Red	Green	Blue
The proposed algorithm	Lena	7.2933	7.5812	7.0856	7.9895	7.9894	7.9894
	Peppers	7.3319	7.5242	7.0793	7.9898	7.9886	7.9894
	Baboon	7.7007	7.5129	7.7655	7.9899	7.9896	7.9888
	Panda	7.7118	7.6278	7.7939	7.9888	7.9899	7.9895
	Vegetables	7.7971	7.8215	7.3593	7.9898	7.9895	7.9895
	Average	7.5670	7.6135	7.4167	7.9896	7.9894	7.9893
Ref. [24]	Lena				7.9874	7.9872	7.9866
Ref. [15]	Lena				7.9278	7.9744	7.9705
Ref. [35]	Lena				7.9732	7.9750	7.9715
Ref. [21]	Lena				7.9791	7.9802	7.9827
(bit-level permuted)							
Ref. [21]	Lena				7.9871	7.9881	7.9878
(Using Chen system)							

to discover the meaningful relationship between the original and encrypted images through making a tiny change in the original image to observe difference between two cipher-images. Then the cryptosystem is cracked by tracing such difference. This type of attack is named as differential attack. In order to test the effect of changing a single pixel in the plain-image on the cipher-image, two common measures, i.e., Number of Pixels Change Rate (*NPCR*) and Unified Average Changing Intensity (*UACI*) [35], are introduced. We can compute  $NPCR_{R,G,B}$  and  $UACI_{R,G,B}$  of the color images according to the following formulae:

$$NPCR_{R,G,B} = \frac{\sum_{i=1}^M \sum_{j=1}^N D_{R,G,B}(i, j)}{M \times N} \times 100 \tag{28}$$

$$UACI_{R,G,B} = \frac{\sum_{i=1}^M \sum_{j=1}^N |C'_{R,G,B}(i, j) - C_{R,G,B}(i, j)| / 255}{M \times N} \times 100 \tag{29}$$

where  $M$  and  $N$  are the width and the height of the image, respectively;  $C_{R,G,B}$  and  $C'_{R,G,B}$  are the two cipher-images and their corresponding plain-images have only one pixel difference;  $D_{R,G,B}(i, j)$  is defined as

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

In order to test our proposed method, the original plain-image is firstly ciphered using the given secret keys. Next, one pixel in the original image is arbitrarily chosen and changed slightly. And the modified image is encrypted again using the same secret keys. Finally, two cipher-images are utilized to obtain  $NPCR_{R,G,B}$  and  $UACI_{R,G,B}$ . This kind of test is executed over 150 times with diverse color images. Table 7 gives the results of  $NPCR_{R,G,B}$  and  $UACI_{R,G,B}$  by using the presented algorithm. As can be seen from the results, the average values of  $NPCR_{R,G,B}$  and  $UACI_{R,G,B}$  are 99.6118% (larger than the expected value 99.6094%) and 33.4500% (very close to the expected value 33.4635%), respectively. So the proposed scheme is resistant to the

**Table 7** Average  $NPCR_{R,G,B}$  and  $UACI_{R,G,B}$  of different color images using the presented algorithm

Test image	Average $NPCR_{R,G,B}$ (%)			Average $UACI_{R,G,B}$ (%)		
	Red	Green	Blue	Red	Green	Blue
Lena	99.6052	99.6060	99.6113	33.4280	33.4966	33.3779
Peppers	99.6060	99.6286	99.5874	33.4959	33.4874	33.4302
Baboon	99.6024	99.6252	99.6004	33.4311	33.4500	33.4935
Panda	99.6062	99.5962	99.6220	33.4335	33.4903	33.5158
Vegetables	99.6177	99.6162	99.6458	33.4517	33.2521	33.5158
Average for each component	99.6075	99.6144	99.6134	33.4480	33.4353	33.4666
Average for all images	99.6118			33.4500		

differential attacks. Furthermore, we compare the  $NPCR$  and  $UACI$  results on the Lena image using our proposed scheme with some other algorithms, as displayed in Table 8. Obviously, the performance of the proposed algorithm is superior to those methods in [5, 14, 23, 29, 35, 46], and it has a better ability to resist the differential attacks.

#### 4.6 Mean absolute error (MAE)

Mean absolute error ( $MAE$ ) is introduced to measure the difference between the plain-image and the cipher-image [30]. The larger the value of  $MAE$ , the more secure the encryption scheme is.  $MAE$  is defined as

$$MAE_{R,G,B} = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N |C_{R,G,B}(i,j) - P_{R,G,B}(i,j)| \quad (30)$$

where  $P$  and  $C$  are the plain-image and the cipher-image, respectively.  $M$  and  $N$  are the width and the height of the image, respectively.

The results for  $MAE$  using the proposed encryption algorithm and the encryption method in [31] are provided in Table 9. The large values of  $MAE$  demonstrate that the cipher-images are highly different from their corresponding plain-images, which again confirms that the security of the presented method is high.

#### 4.7 Resistance to known plaintext attack and chosen plaintext attack

In our proposed encryption method, the key stream  $K$  depend on not only the secret keys (the initial conditions and the parameters of OCML), but the original plain-image. So the key

**Table 8** Comparison of the average  $NPCR$  and  $UACI$  on the Lena image by using different algorithms

Algorithm	Average $NPCR$ (%)	Average $UACI_{R,G,B}$ (%)
The proposed algorithm	99.6075	33.4342
Ref. [46]	99.2173	33.4055
Ref. [14]	99.52	26.7933
Ref. [35]	99.5843	33.3755
Ref. [29]	99.59	17.605
Ref. [23]	99.6017	28.1370
Ref. [5]	99.2938	33.1803

**Table 9** The MAE values for different color images

Test images	MAE		
	Red	Green	Blue
Lena	84.4865	77.8223	70.4894
Peppers	73.8372	86.6486	86.3907
Baboon	76.3557	72.9336	80.0563
Panda	84.5212	85.5103	85.9122
Vegetables	89.9734	85.9789	89.4403
Average for each component	81.8348	81.7787	82.4578
Average for all images	82.0238		
Ref. [31]	80.2		

stream for image diffusion is changeable when different plain-images are encrypted, even though the same secret keys are employed. In addition, different key stream will be obtained if choosing distinct plain-image to encrypt. The attacker cannot extract useful information by encrypting some special images because the resultant information is related to those chosen-images. Therefore, the attacks proposed in [18, 22, 53] become ineffective on our presented cryptosystem. The proposed encryption scheme can well withstand the known-plaintext and chosen-plaintext attacks.

#### 4.8 Robustness against some common image processing operations

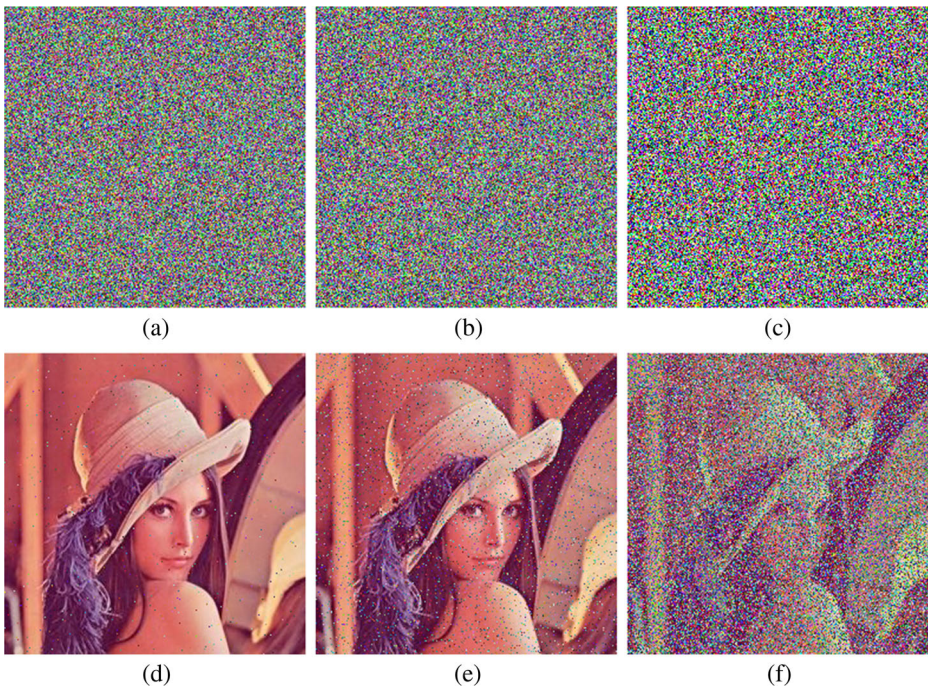
In real applications, the images are inevitably attacked by various image processing operations such as noise addition, cropping, JPEG compression, contrast adjustment, histogram equalization etc. The cryptosystem should be robust so that the plain-image can still be recovered when the cipher-image is attacked. For convenience, in the following simulations, we assume that the encrypted Lena image shown in Fig. 3(e) is attacked by the image processing operations, and then decrypt the attacked cipher-images. The results are described in detail as follows.

##### 4.8.1 Noise addition

As we know, noise is ubiquitous in various applications. In image processing, addition of noise will result in degradation and distortion of the image. In this subsection, the robustness of the proposed encryption algorithm is tested through decrypting the cipher-images contaminated by three kinds of noise, i.e., Pepper & Salt noise, Gaussian noise and multiplicative noise.

**Example I** Fig. 8(a), (b) and (c) show the cipher-images contaminated by Pepper & Salt noise with different noise densities, i.e., 0.005, 0.05 and 0.5, respectively. The proposed scheme is employed to decrypt the noise-contaminated ciphered images Fig. 8(a), (b) and (c). The decrypted images are displayed in Fig. 8(d), (e) and (f), respectively. The corresponding *PSNR* values between the decrypted and original Lena images are 29.03 dB, 19.35 dB and 9.99 dB, respectively.

**Example II** Gaussian white noise with mean value 0 and different variance values is added to the cipher-image in Fig. 3(e). Figure 9(a), (b) and (c) display the cipher-images corrupted by



**Fig. 8** Image decryption as the cipher-image in Fig. 3(e) attacked by salt & pepper noise: **a** Encrypted Lena image under adding Pepper & Salt noise with noise density 0.005; **b** Encrypted Lena image under adding Pepper & Salt noise with noise density 0.05; **c** Encrypted Lena image under adding Pepper & Salt noise with noise density 0.5; **d** Decrypted Lena image from (a); **e** Decrypted Lena image from (b); **f** Decrypted Lena image from (c)

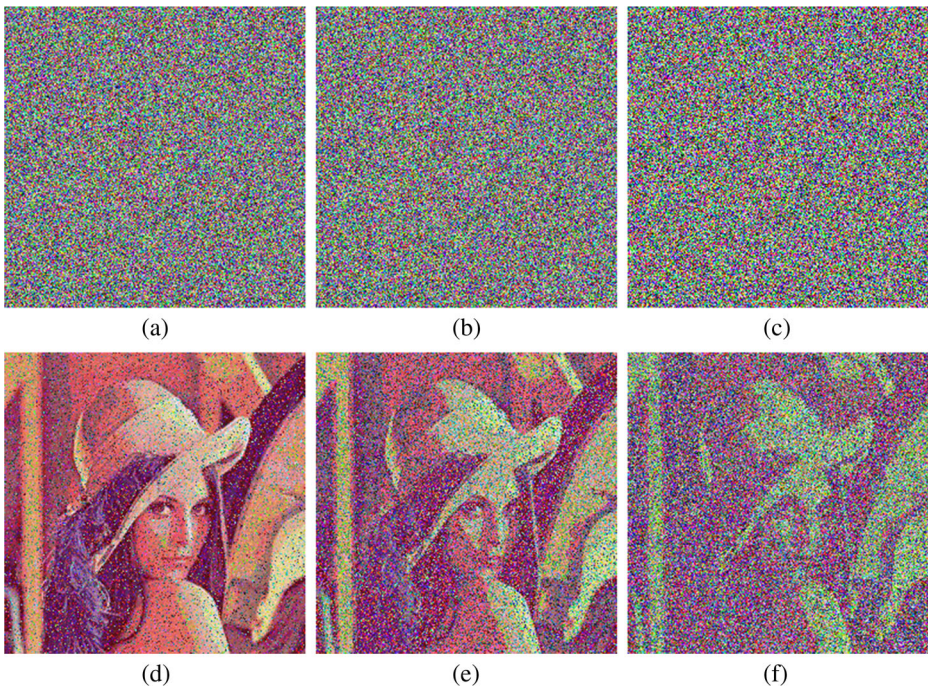
Gaussian white noise with mean value 0 and variance values 0.001, 0.01 and 0.1, respectively. Corresponding decrypted images are shown in Fig. 9(d), (e) and (f). The *PSNR* values between the decrypted and original Lena images are 15.42 dB, 11.46 dB and 9.10 dB, respectively.

**Example III** The cipher-image in Fig. 3(e) is contaminated by multiplicative noise with mean value 0 and different variance values, i.e., 0.001, 0.01 and 0.1. The corrupted encrypted Lena images are depicted separately in Fig. 10(a), (b) and (c). Figure 10(d), (e) and (f) plot the decrypted Lena images. The *PSNR* values between the decrypted and original Lena images are 17.87 dB, 13.48 dB and 9.98 dB, respectively.

From the above results, it can be clearly seen that the noise-contaminated encrypted Lena image can still be decrypted appropriately, i.e., most information can be recovered. For three types of noise, our proposed scheme has the best robustness against Pepper & Salt noise. Furthermore, it becomes more and more difficult to decrypt correctly the noise-corrupted cipher-image with increasing the noise level.

#### 4.8.2 Cropping

Image cropping is a very common operation in image processing. Cropping an image means that some rows or columns of the image are deleted or hidden. So this manipulation will lead to data loss. Figure 11 shows the results of the cropping attacks.

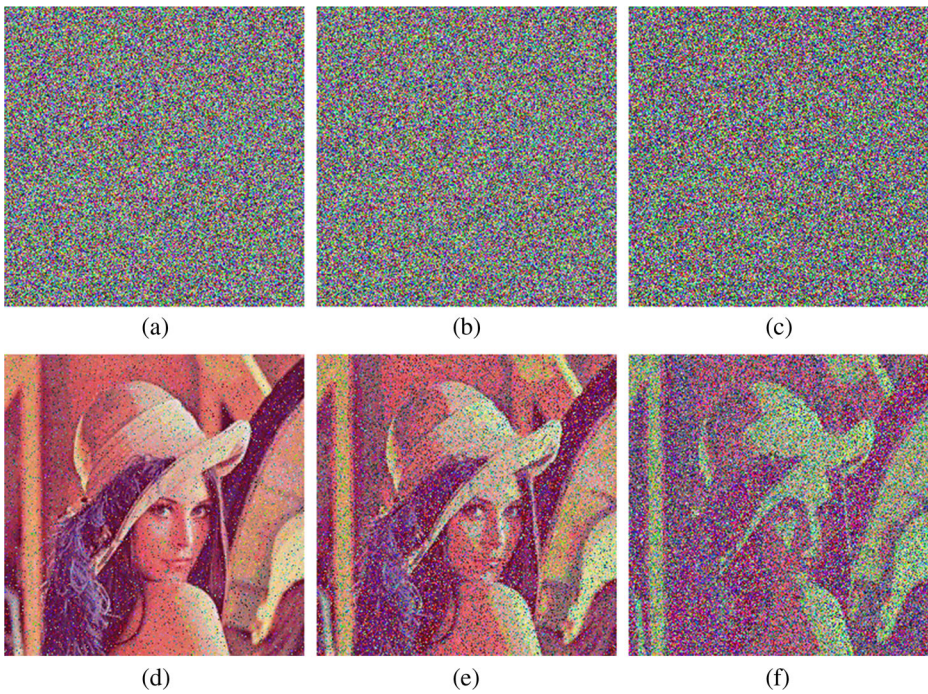


**Fig. 9** Image decryption as the cipher-image in Fig. 3(e) attacked by Gaussian noise: **a** Encrypted Lena image under adding Gaussian noise with mean value 0 and variance value 0.001; **b** Encrypted Lena image under adding Gaussian noise with mean value 0 and variance value 0.01; **c** Encrypted Lena image under adding Gaussian noise with mean value 0 and variance value 0.1; **d** Decrypted Lena image from (a); **e** Decrypted Lena image from (b); **f** Decrypted Lena image from (c)

Figure 11(a) displays the cropped cipher-image, where a data cut with size  $170 \times 80$  in the upper-left of the cipher-image in Fig. 3(e) is deleted. Corresponding decrypted image is shown in Fig. 11(b). A block with size  $95 \times 70$  in the lower-right of the cipher-image in Fig. 3(e) is removed, as displayed in Fig. 11(c). Figure 11(d) plots its corresponding decrypted image. Obviously, the decrypted images still contain most of the original visual information. So our proposed encryption method is robust against the cropping attacks.

#### 4.8.3 JPEG compression

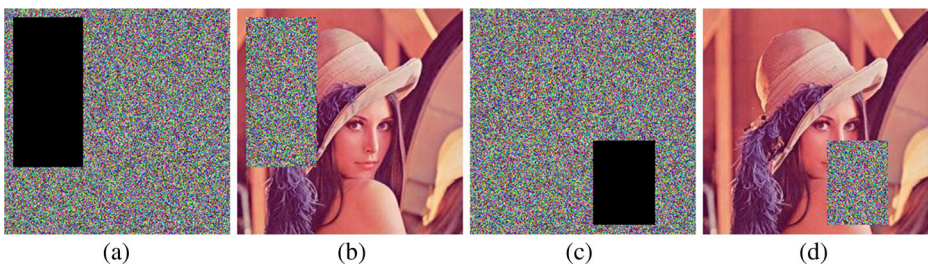
Another common attack in real applications is JPEG compression. In this experiment, the encrypted Lena image in Fig. 3(e) are compressed with different quality factors, i.e., 20, 50 and 80, as displayed in Fig. 12(a), (b) and (c). The quality factor is a kind of measure for JPEG compression, usually in the interval  $[1, 100]$ . The larger the factor, the better the quality of the image after JPEG compression, the smaller the compression rate is. Corresponding decrypted Lena images are separately shown in Fig. 12(d), (e) and (f). The *PSNR* values between the decrypted and original Lena images are 8.83 dB, 9.40 dB and 9.68 dB, respectively. Simulation results show that the decrypted images from the cipher-images after JPEG compression are still recognizable. Thus the presented encryption algorithm is robust against JPEG compression attack.



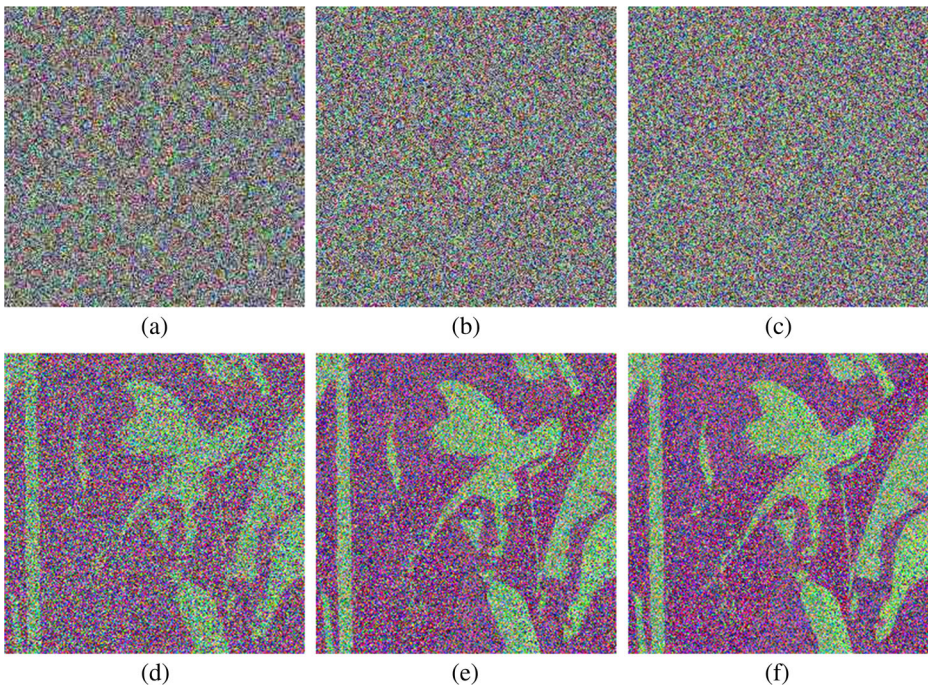
**Fig. 10** Image decryption as the cipher-image in Fig. 3(e) attacked by multiplicative noise: **a** Encrypted Lena image under adding multiplicative noise with mean value 0 and variance value 0.001; **b** Encrypted Lena image under adding multiplicative noise with mean value 0 and variance value 0.01; **c** Encrypted Lena image under adding multiplicative noise with mean value 0 and variance value 0.1; **d** Decrypted Lena image from (a); **e** Decrypted Lena image from (b); **f** Decrypted Lena image from (c)

#### 4.8.4 Contrast adjustment

Contrast is the difference in brightness between objects or regions. In real life, the images may be attacked by adjusting the contrast. The contrast of the encrypted Lena image in Fig. 3(e) is individually increased by 50% and 20%, as shown in Fig. 13(a) and (c). Corresponding decrypted Lena images are displayed in Fig. 13(b) and (d). The *PSNR* value between the decrypted and original Lena images is 10.03 dB and 12.10 dB, respectively. It is evident that most information of the original image can still be recovered. From the results, our proposed scheme is robust against contrast adjustment attacks.



**Fig. 11** Test of image cropping attacks: **a** Encrypted Lena image with a  $170 \times 80$  data cut; **b** Decrypted Lena image from (a); **c** Encrypted Lena image with a  $95 \times 70$  data cut; **d** Decrypted Lena image from (c)

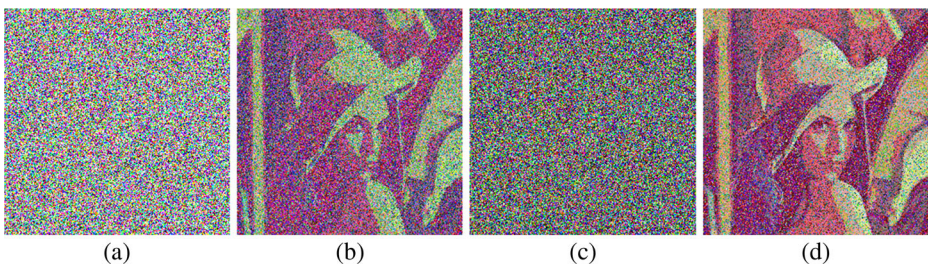


**Fig. 12** Test of JPEG compression attacks: **a** JPEG compressed cipher-image of Lena with quality factor 20; **b** JPEG compressed cipher-image of Lena with quality factor 50; **c** JPEG compressed cipher-image of Lena with quality factor 80; **d** Decrypted Lena image from (a); **e** Decrypted Lena image from (b); **f** Decrypted Lena image from (c)

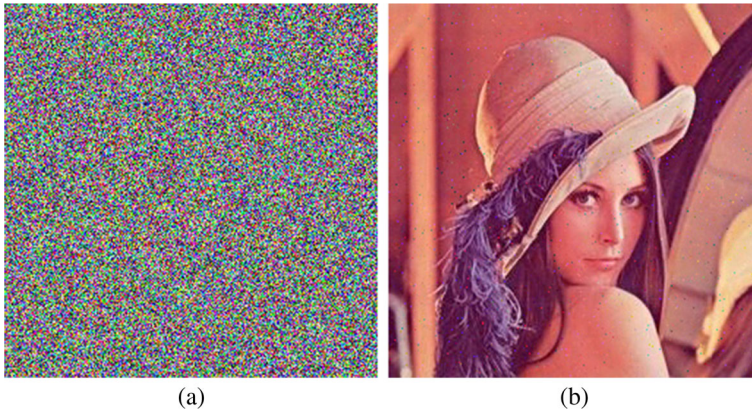
#### 4.8.5 Histogram equalization

Histogram equalization is a technique for adjusting image intensities to enhance contrast. Figure 14(a) shows the encrypted Lena image attacked by histogram equalization. Figure 14(b) displays the corresponding decrypted Lena image. The *PSNR* value between the decrypted and original Lena images is 30.91 dB. As can be seen, the original Lena image can be decrypted well when the cipher-image is attacked by histogram equalization.

**Remark 3** It can be found from the above results that the proposed image encryption scheme is robust against some common image processing operations such as noise addition, cropping, JPEG compression, contrast adjustment and histogram equalization.



**Fig. 13** Test of contrast adjustment attacks: **a** Encrypted Lena image attacked by contrast adjustment (increased by 50%); **b** Decrypted Lena image from (a); **c** Encrypted Lena image attacked by contrast adjustment (increased by 20%); **d** Decrypted Lena image from (c)



**Fig. 14** Test of histogram equalization attacks: **a** Encrypted Lena image attacked by histogram equalization; **b** Decrypted Lena image from (a)

#### 4.9 Computational complexity analysis

The computational complexity of an encryption algorithm will significantly influence the speed performance. To evaluate the computational complexity of the proposed algorithm, the time complexity in each step is given as follows. The image matrix decomposition occurs in the first step. Hence the time complexity is  $O(3 \times M \times N)$  where  $M$  and  $N$  are the dimensions of a RGB image. In Step 2, the time-consuming part is the number of the floating-point operations for CML. Corresponding time complexity is  $O(3 \times M \times N)$ . Step 3 is to transform the pixel matrices into the DNA sequence matrices. So the time complexity in Step 3 is  $O(12 \times M \times N)$ . The time complexity in Step 4 is  $O(12 \times M \times N)$  since the time-consuming part is the number of the DNA XOR operations. Similar to Step 4, the time complexity in Step 5 is also  $O(12 \times M \times N)$ . The time-consuming part in Step 6 is the number of the transformation operations for the DNA sequence matrices, where the time complexity is  $O(24 \times M \times N)$ . In the last stage, three key streams with size of  $M \times N$  are used to modify the image pixels. Thus the time complexity is  $O(3 \times M \times N)$ . Therefore, the total time complexity of the presented encryption scheme is  $O(24 \times M \times N)$ . However, our proposed algorithm can be executed in a parallel mode, which can accelerate the operation speed. Considering the ultra-large-scale parallel computing power of DNA computing, the time cost is indeed negligible.

#### 5 Conclusions

This paper proposes a robust and lossless color image encryption scheme based on DNA computing and OCML. First of all, the original color image is randomly encoded by the DNA encoding rules and three DNA matrices are obtained. Then we implement the DNA XOR operation on the DNA matrices to encrypt the plain-image. And the DNA matrices are transformed into three gray images, i.e., the red, green and blue components, according to the DNA decoding rules. Finally, to further strengthen the security of the cryptosystem, the pixel values of the above image are modified by a key stream generated from both OCML and the plain-image. Numerical experiments and performance analysis have shown the superior encryption effect and high security of the presented image encryption algorithm. Moreover,



our proposed scheme is robust against some common image processing operations such as noise adding, cropping, JPEG compression, contrast adjustment, histogram equalization etc.

**Acknowledgements** This research was jointly supported by the National Natural Science Foundation of China (Grant Nos 61004006 and 61203094), China Postdoctoral Science Foundation (Grant Nos 2013 M530181 and 2015 T80396), Program for Science & Technology Innovation Talents in Universities of Henan Province, China (Grant No 14HASTIT042), the Foundation for University Young Key Teacher Program of Henan Province, China (Grant No 2011GGJS-025), Shanghai Postdoctoral Scientific Program (Grant No 13R21410600).

## References

1. Adleman L (1994) Molecular computation of solutions of combinational problems. *Science* 266:1021–1024
2. Benham CJ, Mielke SP (2005) DNA mechanics. *Annu Rev Biomed Eng* 7:21–53
3. Celand CT, Risca V, Bancroft C (1999) Hiding messages in DNA microdots. *Nature* 399:533–534
4. Chen G, Mao Y, Chui C (2004) A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Soliton Fract* 21(3):749–761
5. Enayatifar R, Abdullah A, Isnin IF (2014) Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. *Opt Lasers Eng* 56:83–93
6. Fridrich J (1998) Symmetric ciphers based on two-dimensional chaotic maps. *Int J Bifurcat Chaos* 8(6):1259–1284
7. Gao T, Chen Z (2008) A new image encryption algorithm based on hyper-chaos. *Phys Lett A* 372(4):394–400
8. Ge X, Liu F, Lu B, Wang W (2011) Cryptanalysis of a spatiotemporal chaotic image/video cryptosystem and its improved version. *Phys Lett A* 375(5):908–913
9. A. Gehani, T.H. LaBean, J.H. Reif, DNA-based cryptography, *Dimacs Series in Discrete Mathematics and Theoretical Computer Science* 54 (2000) 233–249
10. Gu G, Ling J (2014) A fast image encryption method by using chaotic 3D cat maps. *Optik* 125(17):4700–4705
11. Head T, Rozenberg G, Bladergroen RS, Breek CKD, Lommerse PHM, Spaink HP (2000) Computing with DNA by operating on plasmids. *Biosystems* 57(2):87–93
12. Hermassi H, Rhouma R, Belghith S (2012) Security analysis of image cryptosystems only or partially based on a chaotic permutation. *J Syst Softw* 85(9):2133–2144
13. Hua Z, Zhou Y, Pun CM, Chen CLP (2015) 2D sine logistic modulation map for image encryption. *Inf Sci* 297:80–94
14. Huang CK, Nien HH (2009) Multi chaotic systems based pixel shuffle for image encryption. *Opt Commun* 282(11):2123–2127
15. Kadir A, Hamdulla A, Guo W (2014) Color image encryption using skew tent map and hyper chaotic system of 6th-order CNN. *Optik* 125(5):1671–1675
16. Kocaeriv L (2001) Chaos-based cryptography: a brief overview. *IEEE Circuits Syst Mag* 1(3):6–21
17. Li P, Li Z, Halang WA, Chen G (2005) A multiple pseudorandom-bit generator based on a spatiotemporal chaotic map. *Phys Lett A* 349(6):467–473
18. Li C, Li S, Chen G, Halang WA (2009) Cryptanalysis of an image encryption scheme based on a compound chaotic sequence. *Image Vis Comput* 27(8):1035–1039
19. Li C, Zhang LY, Ou R, Wong KW, Shu S (2012) Breaking a novel colour image encryption algorithm based on chaos. *Nonlinear Dyn* 70(4):2383–2388
20. Lian S, Sun J, Wang Z (2005) A block cipher based on a suitable use of chaotic standard map. *Chaos Soliton Fract* 26(1):117–129
21. Liu H, Wang X (2011) Color image encryption using spatial bit level permutation and high-dimension chaotic system. *Opt Commun* 284(16–17):3895–3903
22. Liu L, Zhang Q, Wei X (2012) A RGB image encryption algorithm based on DNA encoding and chaos map. *Comput Electr Eng* 38:1240–1248
23. Liu H, Wang X, Kadir A (2012) Image encryption using DNA complementary rule and chaotic maps. *Appl Soft Comput* 12(5):1457–1466
24. Liu H, Wang X, Kadir A (2013) Color image encryption using Choquet fuzzy integral and hyper chaotic system. *Optik* 124(18):3527–3533
25. Liu Y, Tang J, Xie T (2014) Cryptanalyzing a RGB image encryption algorithm based on DNA encoding and chaos map. *Opt Laser Technol* 60:111–115
26. Mandelkern M, Elias JG, Eden D, Crothers DM (1981) The dimensions of DNA in solution. *J Mol Biol* 152(1):153–161
27. Matthews R (1989) On the derivation of a chaotic encryption algorithm. *Cryptologia* 13:29–42
28. May R (1976) Simple mathematical models with very complicated dynamics. *Nature* 261:459–467

29. Ning K (2009) A pseudo DNA cryptography method (PDF Download Available). Available from: [https://www.researchgate.net/publication/24164703\\_A\\_Pseudo\\_DNA\\_Cryptography\\_Method](https://www.researchgate.net/publication/24164703_A_Pseudo_DNA_Cryptography_Method). Accessed 4 June 2017
30. Norouzi B, Mirzakhaki S (2014) A fast color image encryption algorithm based on hyper-chaotic systems. *Nonlinear Dyn* 78(2):995–1015
31. Norouzi B, Seyedzadeh SM, Mirzakhaki S, Mosavi MR (2013) A novel image encryption based on hash function with only two-round diffusion process. *Multimedia Systems* 20(1):45–64
32. Özkaynak F, Özer AB, Yavuz S (2012) Cryptanalysis of a novel image encryption scheme based on improved hyperchaotic sequences. *Opt Commun* 285(24):4946–4948
33. Pareek NK, Patidar V, Sud KK (2006) Image encryption using chaotic logistic map. *Image Vis Comput* 24(9):926–934
34. Rhouma R, Belghith S (2008) Cryptanalysis of a new image encryption algorithm based on hyper-chaos. *Phys Lett A* 372(38):5973–5978
35. Rhouma R, Meherzi S, Belghith S (2009) OCML-based colour image encryption. *Chaos Soliton Fract* 40(1):309–318
36. Shyam M, Kiran N, Maheswaran V (2007) A novel encryption scheme based on DNA computing. In: *Proceedings of the 14th IEEE International Conference on High Performance Computing, HiPC'2007*. IEEE, New York, USA
37. Solak E, Rhouma R, Belghith S (2010) Cryptanalysis of a multi-chaotic systems based image cryptosystem. *Opt Commun* 283(2):232–236
38. Sui L, Gao B (2013) Color image encryption based on gyration transform and Arnold transform. *Opt Laser Technol* 48:530–538
39. Teng L, Wang X (2012) A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive. *Opt Commun* 285(20):4048–4054
40. Tong X, Cui M (2009) Image encryption scheme based on 3D baker with dynamical compound chaotic sequence cipher generator. *Signal Process* 89(4):480–491
41. Wang Z, Bovik AC (2006) *Modern image quality assessment, synthesis lectures on image, Video & Multimedia Processing*. Morgan & Claypool, San Rafael
42. Wang Y, Wong K, Liao X, Chen G (2011) A new chaos-based fast image encryption algorithm. *Appl Soft Comput* 11(1):514–522
43. Wang X, Teng L, Qin X (2012) A novel colour image encryption algorithm based on chaos. *Signal Process* 92(4):1101–1108
44. Wang W, Tan H, Pang Y, Li Z, Ran P, Wu J (2016) A novel encryption algorithm based on DWT and multichaos mapping. *J Sens* 2016(2646205):1–7
45. Watson JD, Crick FHC (1953) A structure for deoxyribose nucleic acid. *Nature* 171:737–738
46. Wei X, Guo L, Zhang Q, Zhang J, Lian S (2012) A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system. *J Syst Softw* 85(2):290–299
47. Xiao GZ, Lu MX, Qin L, Lai XJ (2006) New field of cryptography: DNA cryptography. *Chin Sci Bull* 51(12):1413–1420
48. Ye G (2010) Image scrambling encryption algorithm of pixel bit based on chaos map. *Pattern Recogn Lett* 31(5):347–354
49. Zhang G, Liu Q (2011) A novel image encryption method based on total shuffling scheme. *Opt Commun* 284(12):2775–2780
50. Zhang Y, Wang X (2014) A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice. *Inf Sci* 273:329–351
51. Zhang Q, Wei X (2013) A novel couple images encryption algorithm based on DNA subsequence operation and chaotic system. *Optik* 124(23):6276–6281
52. Zhang Q, Guo L, Wei X (2010) Image encryption using DNA addition combining with chaotic maps. *Math Comput Model* 52(11–12):2028–2035
53. Zhang Q, Guo L, Wei X (2013) A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. *Optik* 124(18):3596–3600
54. Zhang Q, Liu L, Wei X (2014) Improved algorithm for image encryption based on DNA encoding and multi-chaotic maps. *AEU-Int J Electron Commun* 68(3):186–192
55. Zhang Y, Wen W, Su M, Li M (2014) Cryptanalyzing a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. *Optik* 125(4):1562–1564
56. Zheng XD, Xu J, Li W (2009) Parallel DNA arithmetic operation based on n-moduli set. *Appl Math Comput* 212(1):177–184
57. Zhou Y, Bao L, Chen CLP (2014) A new 1D chaotic system for image encryption. *Signal Process* 97:172–182
58. Zhu C (2012) A novel image encryption scheme based on improved hyperchaotic sequences. *Opt Commun* 285(1):29–37
59. Zhu Z, Zhang W, Wong K, Yu H (2011) A chaos-based symmetric image encryption scheme using a bit-level permutation. *Inf Sci* 181:1171–1186



**Xiangjun Wu** received the B.S. degree in Department of Computer and Information Engineering in 2003 from Henan university, Kaifeng, Henan Province, China, M.S. degree from the School of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, Liaoning Province, China, in 2006, and Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2011. He is currently a full professor and vice dean at College of Software, Henan University. His current research interests include nonlinear dynamics and control, complex networks and systems, cryptography and secure communications. Prof. Wu is a member of the editorial boards of several international scientific journals. He is also a reviewer of more than 40 international journals.



**Jürgen Kurths** studied mathematics at the University of Rostock, Rostock, Germany, and received the Ph.D. degree from the GDR Academy of Sciences, erlin, Germany, in 1983. He was a Full Professor at the University of Potsdam, Potsdam, Germany, from 1994 to 2008. Since 2008, he has been a Professor of nonlinear dynamics at the Humboldt University, Berlin, and the Chair of the research domain Transdisciplinary Concepts of the Potsdam Institute for Climate Impact Research, Potsdam. Since 2009, he has been the sixth Century Chair of Aberdeen University, Aberdeen, U.K. He has published more than 600 papers (H-factor: 60). His current research interests include synchronization, complex networks, and time series analysis and their applications. Dr. Kurths is a fellow of the American Physical Society. He received the Alexander von Humboldt Research Award from CSIR, India, in 2005, and the Honorary Doctorate from the Lobachevsky University Nizhny Novgorod, in 2008, and from the State University Saratov, in 2012. He became a member of the Academia Europaea in 2010 and of the Macedonian Academy of Sciences and Arts in 2012. He is an Editor of PLoS ONE, the Philosophical Transaction of The Royal Society A, and the Journal of Nonlinear Science, CHAOS.



**Haibin Kan** received the Ph.D. degree from Fudan University, Shanghai, China, in 1999. After receiving the Ph.D. degree, he became a faculty of Fudan University. From June 2002 to December 2005, he was with the Japan Advanced Institute of Science and Technology as an Assistant Professor. He went back Fudan University in 2006, where he is currently a Full Professor. His research topics include coding theory, communication and information security.