ISSN 1674-1056
http://cpb.iphy.ac.cn
http://iopscience.iop.org/cpb
中国物理 B
Chinese Physics
中国物理 Chinese Physics B

New chaotical image encryption algorithm based on Fisher-Yatess scrambling and DNA coding

Xing-Yuan Wang(王兴元), Jun-Jian Zhang(张钧荐), Fu-Chen Zhang(张付臣), Guang-Hui Cao(曹光辉)

Citation: Chin. Phys. B . 2019, 28(4): 040504. **doi:** 10.1088/1674-1056/28/4/040504
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

A novel pseudo-random coupled LP spatiotemporal chaos and its application in image encryption

Xingyuan Wang(王兴元), Yu Wang(王宇), Siwei Wang(王思伟), Yingqian Zhang(张盈谦), Xiangjun Wu(武相军)
Chin. Phys. B . 2018, 27(11): 110502. **doi:** 10.1088/1674-1056/27/11/110502

Image encryption technique based on new two-dimensional fractional-order discrete chaotic map and Menezes-Vanstone elliptic curve cryptosystem

Zeyu Liu(刘泽宇), Tiecheng Xia(夏铁成), Jinbo Wang(王金波)
Chin. Phys. B . 2018, 27(3): 030502. **doi:** 10.1088/1674-1056/27/3/030502

A novel color image encryption scheme using fractional-order hyperchaotic system and DNA sequence operations

Li-Min Zhang(张立民), Ke-Hui Sun(孙克辉), Wen-Hao Liu(刘文浩), Shao-Bo He(贺少波)
Chin. Phys. B . 2017, 26(10): 100504. **doi:** 10.1088/1674-1056/26/10/100504

An image encryption scheme based on three-dimensional Brownian motion and chaotic system

Xiu-Li Chai(柴秀丽), Zhi-Hua Gan(甘志华), Ke Yuan(袁科), Yang Lu(路杨), Yi-Ran Chen(陈怡然)
Chin. Phys. B . 2017, 26(2): 020504. **doi:** 10.1088/1674-1056/26/2/020504

A self-cited pixel summation based image encryption algorithm

Guo-Dong Ye(叶国栋), Xiao-Ling Huang(黄小玲), Leo Yu Zhang(张愉), Zheng-Xia Wang(王政霞)
Chin. Phys. B . 2017, 26(1): 010501. **doi:** 10.1088/1674-1056/26/1/010501

New chaotical image encryption algorithm based on Fisher–Yatess scrambling and DNA coding*

Xing-Yuan Wang(王兴元)[1,2,†], Jun-Jian Zhang(张钧荐)[2,‡], Fu-Chen Zhang(张付臣)[3], and Guang-Hui Cao(曹光辉)[4]

[1] School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China
[2] Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China
[3] College of Mathematics and Statistics, Chongqing Technology and Business University, Chongqing 400067, China
[4] School of Electronics & Information Engineering, Liaoning University of Technology, Jinzhou 121001 China

(Received 29 October 2018; revised manuscript received 10 February 2019; published online 5 March 2019)

Based on the Fisher–Yatess scrambling and DNA coding technology, a chaotical image encryption method is proposed. First, the SHA-3 algorithm is used to calculate the hash value of the initial password, which is used as the initial value of the chaotic system. Second, the chaotic sequence and Fisher–Yatess scrambling are used to scramble the plaintext, and a sorting scrambling algorithm is used for secondary scrambling. Then, the chaotic sequence and DNA coding rules are used to change the plaintext pixel values, which makes the ciphertext more random and resistant to attacks, and thus ensures that the encrypted ciphertext is more secure. Finally, we add plaintext statistics for pixel-level diffusion to ensure plaintext sensitivity. The experimental results and security analysis show that the new algorithm has a good encryption effect and speed, and can also resist common attacks.

Keywords: Fisher–Yatess scrambling, chaotic sequence, DNA coding, image encryption

PACS: 05.45.Ac, 05.45.–a **DOI:** 10.1088/1674-1056/28/4/040504

1. Introduction

With the rapid development of computer technology and internet technology, multimedia information transmission becomes increasingly frequent and necessary. However, the openness and sharing of network environment is a huge threat to the security of image transmission. Information encryption technology is necessary to ensure information security. For image encryption, due to its high redundancy, large data capacity, and strong correlation between information, the requirements for image encryption algorithms are more stringent.[1–3] And the traditional encryption methods, such as DES, AES, and RSA,[4,5] can no longer meet the current image encryption requirements. However, these methods are too time consuming. For real-time image encryption, only fast and secure encryption algorithms are available. In recent years, scholars have proposed some new image encryption algorithms, of which the encryption scheme based on chaos theory should be the most prominent.[6–9] Chaotic systems have good pseudo-random characteristics, and some properties such as sensitivity to initial values and unpredictability of orbits are consistent with cryptographic requirements. Chaotic cryptography has been received wide attention and has been applied to image encryption.[10–12] The method proposed in this paper is also based on chaos theory.

To improve the efficiency of encryption, in this paper a low-dimensional chaotic system and is used and it combines with other methods to solve the problems in some existing methods.

Image encryption algorithms are generally classified as scrambling, replacement, and diffusion. The scrambling process changes the position of the pixel, the replacement process changes the value of the pixel, and the diffusion process spreads the image portion information to the full text range. Many scholars have proposed excellent scrambling algorithms, such as Arnold transform, Baker transform, geometric transform, E-curve transform, *etc*.[13–15] These classical methods have greatly promoted the research in the field of image encryption, but some problems have been exposed in later research. For example, the Arnold transform and the Baker transform have obvious periodicity, and some methods have problems such as poor global scrambling.[16–18] The Fisher–Yatess scrambling method used in this paper is a shuffling algorithm. The advantage is that it does not have any regularity in completely random sorting. Compared with the similar methods used in scrambling, like Arnold transformation which has a finite period, this method is safe. We have not seen anyone use this method for scrambling in similar algorithms, so we use it in this article. Comparing with the original method,

*Project supported by the National Natural Science Foundation of China (Grant Nos. 61173183, 61672124, 61370145, and 11501064), the Password Theory Project of the 13th Five-Year Plan National Cryptography Development Fund, China (Grant No. MMJJ20170203), the China Postdoctoral Science Foundation (Grant No. 2016M590850), and the Scientific and Technological Research Program of Chongqing Municipal Education Commission, China (Grant No. KJ1500605).

†Corresponding author. E-mail: wangxy@dlut.edu.cn
‡Corresponding author. E-mail: 463182719@qq.com

© 2019 Chinese Physical Society and IOP Publishing Ltd

we utilize a chaotic sequence instead of a completely random sequence. Chaotic sequences are pseudo-random. That makes the process reversible.

Scientists have found that the sequence of nucleic acids has a natural quaternary combination, similar to the binary system formed by the semiconductor's on and off. The huge parallelism, ultrahigh storage density and ultra-low energy consumption of DNA are being developed for molecular computing, data storage, cryptography and other fields, which may eventually lead to the birth of new computers, new data storage and new cryptography systems, triggering a new information revolution.[19–22] In 1999, Gehani *et al.* proposed a one-time pad (OTP) mechanism based on DNA and presented two kinds of OTP cryptography schemes of the substitution method and the exclusive OR (xor) method.[23] In 2003, Chen *et al.* constructed a cryptosystem based on DNA molecular sequences.[24] In 2005, Kazuo *et al.* used DNA to solve the problem of key distribution.[25] In 2009, Mousa *et al.* designed an information hiding scheme by using a contrast mapping method to embed ciphertext information into any part of nucleic acid sequence without changing the function of nucleic acid.[26] In recent years, DNA coding methods have been continuously developed, and the forms of use have become diverse.[27] In this paper, we use the chaotic sequence to encode and decode DNA sequence and perform DNA-level diffusion. Due to the nature of DNA coding, the entire replacement process takes a short time and the encryption effect is good.

The target of this paper is to propose a safe and efficient image encryption algorithm. Safety is the lower limit and efficiency is the target. The Fisher–Yatess method is used in scrambling. One scramble can achieve the desired effect. We use a simplified DNA coding to replace the plaintext, and also use a low-dimensional Logistic map. All of this gives us a great advantage in terms of encryption speed. The rest of this paper is organized as follows. The relevant methods are given in Section 2. In Section 3, the encryption method is described. The experimental results and security analysis are presented in Section 4. Finally, some conclusions are drawn in Section 5.

2. Relevant methods

2.1. Fisher–Yatess scrambling

The Fisher–Yatess random scrambling algorithm is also referred to as the Knuth algorithm, which is a classic shuffle algorithm. The essence of the algorithm is to generate a finite set of random arrangements. This algorithm is unbiased with a total of $n!$ possibilities, and each permutation is of equal probability. This algorithm is very efficient, time complexity is $O(n)$, and no extra storage is required. The algorithm is described as follows.

For sequence $A[n]$, perform the following steps:
(i) let $i = n$;
(ii) generate a random integer j of 1 to i;
(iii) exchange A_i, A_j values and let $i = i - 1$;
(iv) repeat steps (ii) and (iii) until $i = 1$.

2.2. Chaotic system

In this article we take one-dimensional chaotic system logistic map for example. One-dimensional logistic map is a very simple chaotic map in terms of mathematical form. It has an extremely complex dynamic behavior and has a wide range of applications in the field of secure communication. Logistic map is calculated as follows:

$$x_{n+1} = \mu x_n(1 - x_n). \tag{1}$$

And $x_n \in (0,1)$, when parameter $\mu \in (3.5699456, 4]$, logistic map is in chaos, and the resulting sequence $\{x_n, n = 0, 1, 2, \ldots\}$ is non-periodic, non-convergent, and very sensitive to initial values. To avoid the periodic window, the parameters in $\mu \in (3.89, 4]$.

2.3. DNA Coding

Deoxyribonucleic acid is a molecular structure composed of four nucleotides. Adenine (A), thymine (T), cytosine (C), and Guanine (G) form the major part of nucleotide. According to the pairing rule of DNA, A and T are paired and C and G are paired. According to the DNA pairing rule rules, there are eight legal combinations, which are shown in Table 1.

Table 1. DNA pairing rules.

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

In the process of encrypting the image, the gray value of each 8-bit pixel can be represented by four encoded DNA sequences. For example, a pixel with a grayscale value of 201 and a binary sequence of 11001001 is encoded according to the DNA rules. We can obtain 8 combinations: TACG, TAGC, ATCG, ATGC, CGTA, CGAT, GCTA, and GCAT. In addition, the algorithm also performs different operations on DNA sequences to encrypt images. The DNA addition, subtraction, and xor operations are listed in Tables 2–4.

Table 2. DNA addition.

+	A	G	C	T
A	A	G	C	T
G	G	C	T	A
C	C	T	A	G
T	T	A	G	C

Table 3. DNA subtraction.

–	A	G	C	T
A	A	T	C	G
G	G	A	T	C
C	C	G	A	T
T	T	C	G	A

Table 4. DNA xor operations.

$\oplus$	A	G	C	T
A	A	G	C	T
G	G	A	T	C
C	C	T	A	G
T	T	C	G	A

3. Encryption algorithm

The encryption algorithm presented in this paper consists of three parts: scrambling, DNA coding and diffusion. For an image P of size $M \times N$, where M is the number of rows and N is the number of columns, the encryption process is described in detail below.

3.1. Generate key and system initial value

Due to the irreversibility and collision resistance of the Hash algorithm, it can effectively defend against known plaintext attacks and select plaintext/ciphertext attacks. In this paper used is the SHA-3 algorithm to generate key and system initial value.[28] Users can input passwords with an uncertain size. The original password is SHA-3 to generate a set of 256-bit hash values K as key. A low-dimensional chaotic systems has the characteristics of high efficiency and high efficiency, and its chaos does not lose in a high-dimensional chaotic system. Therefore, the chaotic sequences in this paper are all generated by using a one-dimensional logistic map. In this paper, the chaotic sequence is used four times. Therefore, four groups of initial values are generated by the key. The odd-numbered and even-numbered sequences of $K$are separated into 128-bit-length sequences K_1 and K_2. Then K_1 and K_2 are divided into bytes, which can be expressed as $K_1 : (k_1, k_2, \ldots, k_{16})$, and $K_2 : (k_{17}, k_{18}, \ldots, k_{32})$. The initial values and control parameters are generated according to the following rules:

$$x = \begin{cases} x_1 = ((k_1 \oplus k_2 \oplus k_3 \oplus k_4)/256 + x_0)/2, \\ x_2 = ((k_5 \oplus k_6 \oplus k_7 \oplus k_8)/256 + x_0)/2, \\ x_3 = ((k_9 \oplus k_{10} \oplus k_{11} \oplus k_{12})/256 + x_0)/2, \\ x_4 = ((k_{13} \oplus k_{14} \oplus k_{15} \oplus k_{16})/256 + x_0)/2, \\ x_5 = ((k_1 \oplus k_3 \oplus k_5 \oplus k_7)/256 + x_0)/2, \\ x_6 = ((k_2 \oplus k_4 \oplus k_6 \oplus k_8)/256 + x_0)/2, \\ x_7 = ((k_9 \oplus k_{11} \oplus k_{13} \oplus k_{15})/256 + x_0)/2, \\ x_8 = ((k_{10} \oplus k_{12} \oplus k_{14} \oplus k_{16})/256 + x_0)/2, \end{cases} \tag{2}$$

$$u = \begin{cases} u_1 = (k_{17} \oplus k_{18} \oplus k_{19} \oplus k_{20})/256, \\ u_2 = (k_{21} \oplus k_{22} \oplus k_{23} \oplus k_{24})/256, \\ u_3 = (k_{25} \oplus k_{26} \oplus k_{27} \oplus k_{28})/256, \\ u_4 = (k_{29} \oplus k_{30} \oplus k_{31} \oplus k_{32})/256, \\ u_5 = (k_{17} \oplus k_{19} \oplus k_{21} \oplus k_{23})/256, \\ u_6 = (k_{18} \oplus k_{20} \oplus k_{22} \oplus k_{24})/256, \\ u_7 = (k_{25} \oplus k_{27} \oplus k_{29} \oplus k_{31})/256, \\ u_8 = (k_{26} \oplus k_{28} \oplus k_{30} \oplus k_{32})/256. \end{cases} \tag{3}$$

Here, x_0 is the control value added to avoid the xor result being 0 and $x_0 \in [0,1]$. In order to avoid the periodic window, the parameters in $\mu \in (3.89, 4]$. The u is converted into μ as follows:

$$\mu = 3.89 + 0.11u. \tag{4}$$

Then the initial value x and parameter μ of each group are brought into the logistic map to generate chaotic sequences.

3.2. Scrambling

Scrambling is divided into two parts. First, the preliminary scrambling is performed through the Fisher–Yatess scrambling algorithm. Image P is converted into a one-dimensional sequence $(p_1, p_2, \ldots, p_{MN})$, x_1 and u_1 are taken into the logistic map and iterate 200 times to remove the transient effect, then a chaotic sequence A_1 of length $M \times N$ is generated. The specific operation is as follows:

i) let $i = M \times N$, repeat steps (ii) and (iii) and after each time $i = i - 1$ until $i = 1$;

ii) $j = \lfloor A_{1_i} \times i \rfloor + 1$;

iii) exchange p_i and p_j values.

After these steps, a preliminary scrambling sequence P' is obtained, and then P' is sorted for scrambling. The x_2 and u_2 are taken into the logistic map, then a chaotic sequence A_2 of length $M \times N$ is generated. A non-repeated sequence B is obtained from the following formula:

$$B_i = \lfloor A_{2_i} \times M \times N \rfloor + i/(M \times N), \tag{5}$$

$$B' = \mathrm{sort}(B). \tag{6}$$

$\mathrm{sort}(\cdot)$ is the sort function. The B' subscript is retained where its element value is the same as B in the sequence D, D is a disorder of the integer sequence from 1 to $M \times N$. Starting to map p_1 to p_{Di} from the first element p_1 of P', exchanging the position from 1 to $M \times N$, then converting it into an image P_1 of size $M \times N$, scrambling is completed here.

3.3. DNA coding and diffusion

Although the scrambled image hides the position information of the pixel, attack methods such as statistical analysis can still obtain a lot of valid information in the plaintext. Therefore, it is necessary to change the pixel value to

hide the plaintext information. Used in this paper is the DNA coding and diffusion with chaotic disturbances to solve this problem. The DNA replacement method used in this paper is divided into three steps. First, the original image is encoded by DNA. Then the DNA sequence after being encoded is used for computing the DNA. Finally, the decoded DNA sequence is decoded to complete the replacement process. The specific method is as follows.

Taking x_3 and u_3 into the logistic map, then generating two chaotic sequence A_3 and A_4 of length $M \times N$, the codec rule sequence R_e and R_d are generated as follows:

$$R_e = \mathrm{mod}\,(\mathrm{unit8}(255 \times A_3), 8), \tag{7}$$

$$R_d = \mathrm{mod}\,(\mathrm{unit8}(255 \times A_4), 8). \tag{8}$$

The pixel value encoding method of the scrambled image P_1 is converted into binary. The unit8 encoding format for grayscale images converts each pixel value into binary followed by eight bits, For each pixel p_i, select its corresponding rule R_{ei} for encoding to generate four-digit characters. The $f(\cdot)$ indicates that the pixel value is encoded according to the corresponding encoding rule, and the full-image encoding is stored as a sequence C of length $4 \times MN$ as expressed below.

$$C_i = f(p_i, R_{\mathrm{ei}}). \tag{9}$$

A line-by-line operation is operated on C to obtain C'. Where $g()$ is any of the reversible DNA algorithm rules. The formula is as follows:

$$C' = \begin{cases} C'_1 = C_1, \\ C'_{i+1} = g(C_{i+1}, C'_i). \end{cases} \tag{10}$$

After the operation, C' is decoded line-by-line according to the decoding rule sequence R_d to generate a binary sequence of length $8 \times MN$, and the resulting sequence is converted into a unit8 encoding format, and then transposed and restored to a grayscale image P_2. Now, the DNA replacement ends here.

To further hide the textual information, the pixel values are diffused after DNA replacement. Take x_4 and u_4 into the logistic map, then a chaotic sequence A_5 of length $M \times N$ will be generated. And diffusion takes place according to the following formula:

$$\begin{cases} S_i = \mathrm{mod}\,(\lfloor A_{5i} \times \lambda \rfloor, 256), \\ T_i = p_i \oplus S_i, \\ T' = \begin{cases} T'_1 = T_1, \\ T'_{i+1} = T_{i+1} \oplus T'_i. \end{cases} \end{cases} \tag{11}$$

Here, p_i is the i-th pixel value of the DNA replacement image P_2. Revert T' into a grayscale image P_3, then it will be the final result of the first round of encryption. The λ is a value calculated from the plaintext P to improve the plaintext sensitivity of the algorithm. Suppose that the image to be encrypted is $P = \{p(i,j)\}$, and $p(i,j)$ is the value at pixel (i,j). We calculate the following values:

$$H_1 = \bigoplus_{i=1}^{M} \bigoplus_{j=1}^{N} p(i,j), \quad H_2 = \sum_{i=1}^{M} \sum_{j=1}^{N} p(i,j) \bmod 256. \tag{12}$$

Here, H_1 is the xor of all gray values of the image, and H_2 is the sum modulo 256 of all pixel values. Let $\lambda = 3000 + H_1 H_2$. In this way, minor modifications to the original image will cause λ to change, and because the algorithm is sensitive to the key, the algorithm will eventually be sensitive to plaintext. It should be pointed out that after the above modification, each image encryption needs to be accompanied by two additional information H_1 and H_2 for decryption. These two pieces of information are stored and transmitted by using the key.

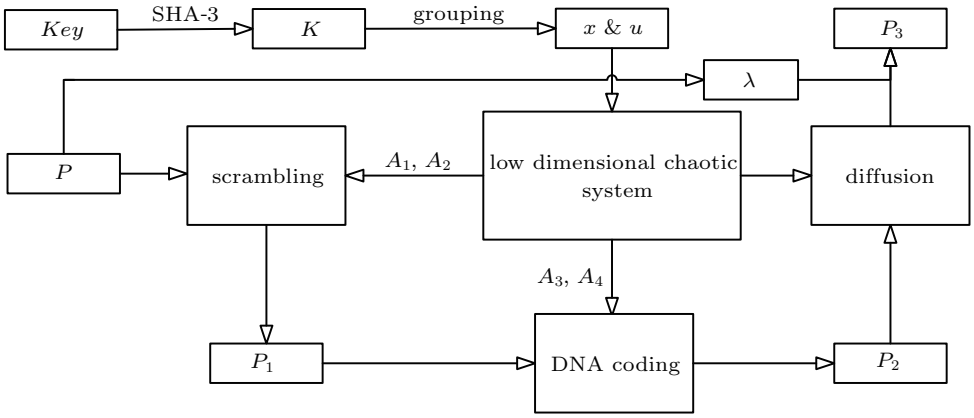


Fig. 1. Encryption process.

We use P_3 as the original image of the second round of encryption. Bring x_5, x_6, x_7, x_8 and u_5, u_6, u_7, u_8 serving as parameters into the algorithm to repeat the above process. Here we use the same λ. The result P_4 of second round of encryption is the result of the entire encryption process. At this point, encryption ends. The encryption process is shown in Fig. 1. First, the SHA-3 algorithm is used to calculate the K. We divide K into groups and we obtain x and u, which are the

initial and control value of the chaotic system. Meanwhile we calculate λ through P. Second, the chaotic sequence (A_1, A_2) and scrambling algorithms are used to scramble the plaintext P. Here we obtain P_1 as an intermediate image. Then, the chaotic sequence (A_3, A_4) and DNA coding rules are used to change the pixel value of P_1, and we obtain P_2. After that, P_2 and chaotic sequences (A_5) and λ are brought into the diffusion algorithm to obtain the first round ciphertext P_3. The second round of encryption is a repetition of the above process.

3.4. Decryption

Decryption is similar to encryption, and it is the reverse of the above steps. First, we calculate $x_1 - x_8$, $u_1 - u_8$, and λ by using K, x_0, H_1, H_2, Eqs. (2), (3), and (12). Encryption consists of two rounds of processes with identical methods and different parameters. We only give a round (from P_3 to P) decryption process here. Specific steps are as follows.

Step 1 Bring x_4 and u_4 into the logistic map, then chaotic sequence A_5 will be generated. And bring A_5 and λ into Eq. (13), then P_2 will be obtained as follows:

$$P_2 = \begin{cases} S_i = \mod(\lfloor A_{5i} \times \lambda \rfloor, 256), \\ T = \begin{cases} T_1 = T'_1, \\ T_{i+1} = T'_i \oplus T'_{i+1}, \end{cases} \\ p_i = S_i \oplus T_i. \end{cases} \tag{13}$$

Step 2 Bring x_3 and u_3 into the logistic map, then two chaotic sequences A_3 and A_4 will be generated. Bring A_3 and A_4 into Eqs. (7) and (8), then R_e and R_d will be obtained. Bring P_2 and R_d into $f()$ just like the encryption process, then C' will be obtained from the following equation:

$$C'_i = f(p_i, R_{\mathrm{di}}). \tag{14}$$

Then C can be calculated from

$$C = \begin{cases} C_1 = C'_1, \\ C_{i+1} = g^{-1}(C_i, C'_{i+1}). \end{cases} \tag{15}$$

Here, $g^{-1}()$ represents the inverse of $g()$. For example, if $g()$ is additive in encryption, then the subtraction is used for decryption. If xor is used for encryption, then the xor is still used for decryption. After the operation, C is decoded line-by-line according to the rule sequence R_e to generate a binary sequence of length $8 \times MN$, and the resulting sequence is converted into a *unit* 8 encoding format and transposed and restored into a grayscale image P_1.

Step 3 Similarly, we bring in x_1, x_2, u_1, u_2 to calculate A_1, A_2, and we obtain B and B' according to Eqs. (5) and (6). In the encryption process, we rearrange P' according to the mapping relationship of B to B' to obtain P_1. Here we restore P_1 into P' according to the mapping of B' to B. The process of restoring P' into P is also the reverse process of Fisher–Yatess scrambling. In the encryption process, we loop to determine the last element of the unsorted part until there is only one element. The decryption process starts from one element. Every loop inserts an element into the correct relative position until all elements return to the correct position, as described below:

(I) let $i = 1$, repeat steps (II) and (III) and after each time $i = i + 1$ until $i = M \times N$;

(II) $j = \lfloor A_{1_{M \times N - i + 1}} \times i \rfloor + 1$;

(III) exchange p_i and p_j values.

4. Experimental results and security analysis

4.1. Experimental results

We chose Lena and Fabio both with a size of 256×256, and pepper to encrypt. Figure 2 shows the results of encrypting the above three grayscale images. After decryption, the plaintext image can be correctly obtained.

4.2. Algorithm complexity analysis

This article uses Matlab 2014 to perform the encryption and decryption program, the computer configuration is 2 GHZ CPU with 4 GB memory and Microsoft Windows 8 operating system. The average time for encrypting a 256×256 pixel grayscale image is 1.15 seconds. When encrypting an image with size $M \times N$, it generates chaotic sequences for a total of $5 \times MN + 200$ operations, scrambling operations for a total of $6 \times MN$ operations, a DNA replacement operation for a total of $12 \times MN$ operations, and a conversion with two further conversions. The diffusion operations will total $2 \times MN$ operations. A total of $25 \times MN$ operations have an algorithm complexity of $O(MN)$.

4.3. Security analysis

4.3.1. Key analysis

Key space To effectively respond to various brute-force attacks, the key space should be large enough. The key in this algorithm consists of $x_1 - x_8$, $u_1 - u_8$, and x_0. The key space is about $256^8 \times 256^8 \times 10e^{15}$. So the key space is about $8 \times 8 \times 2 + 50 = 178$ bits. It is greater than the 100-bit key space security requirement.[29]

Sensitivity analysis of key We set K_1 and $x_0 = 0.3755$ experiment with grayscale image "Lena" of size 256×256. Results are shown in Fig. 3. Figures 3(a)–3(c) show the results of decrypting with the correct key, and figures 3(b) and 3(c) display the results of decryption by using a small difference keys. The values of these keys are shown below:

$K_1 =$ c1a70bcf0a1e031e57ba7bb7a51deb31cff84d79e0f3a5d49afa250f62b512fc,

$K_2 =$ d1a70bcf0a1e031e57ba7bb7a51deb31cff84d79e0f3a5d49afa250f62b512fc,

$K_3 =$ c1a70bcf0a1e031e57ba7bb7a51deb31cff84d79e0f3a5d49afa250f62b512fd.

The experimental results are analyzed and it is found that the original image can be correctly solved only by decrypting the key with the original key, which indicates that the encryption algorithm is sensitive to the key.

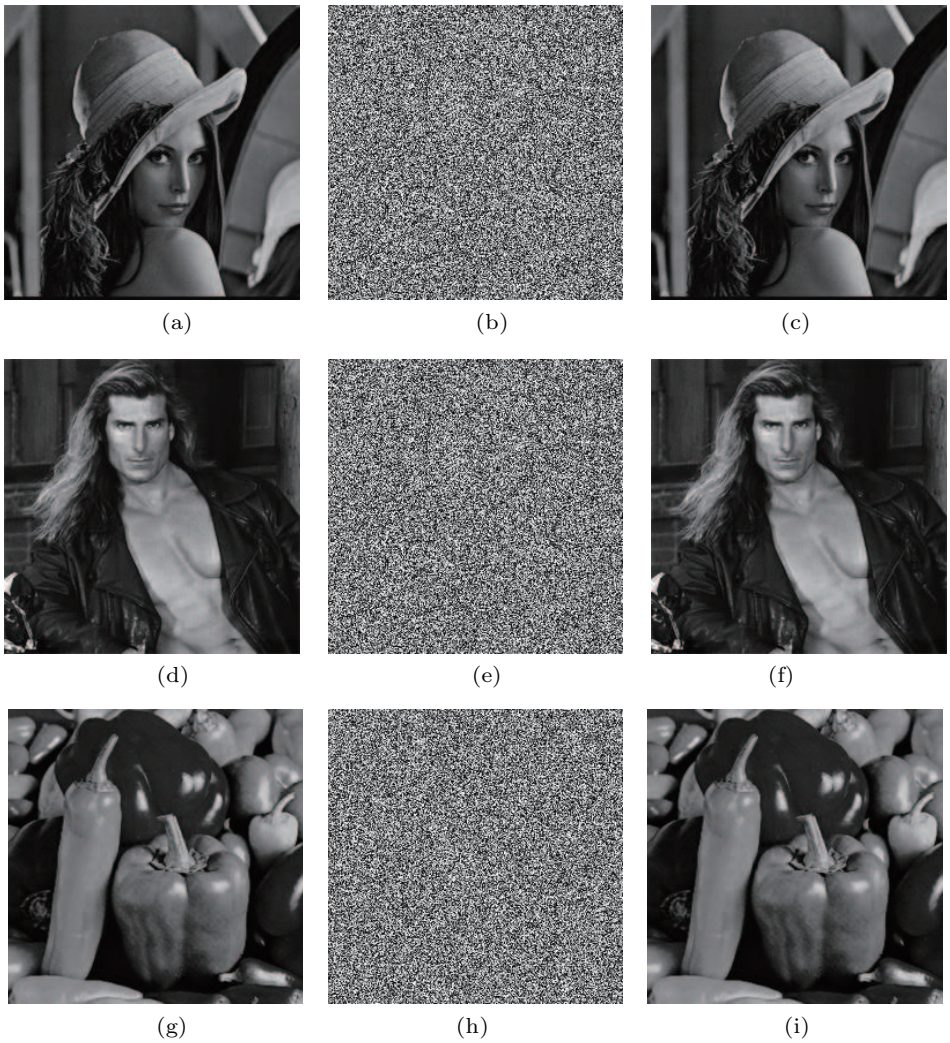


Fig. 2. Experimental results, showing (a) plain image of Lena, (b) encryption image of Lena, (c) decryption image of Lena, (d) plain image of Fabio, (e) encryption image of Fabio, (f) decryption image of Fabio, (g) plain image of Pepper, (h) encryption image of Pepper, and (i) decryption image of Pepper.

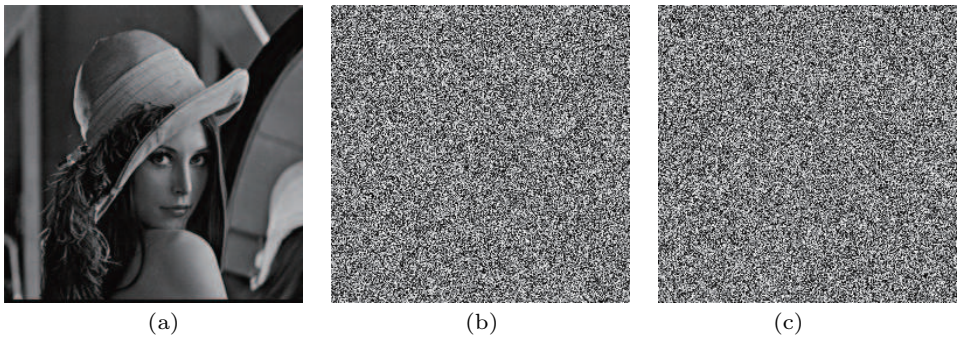


Fig. 3. Decrypt results with correct and incorrect key, showing decryption with (a) key K_1, (b) key K_2, and (c) key K_3.

4.3.2. Histogram analysis

The histogram of ciphertext image is an important feature, showing whether the algorithm can stand the statistical analysis. Histogram describes the distribution of pixel values of an image. If the distribution is uneven, an attacker can obtain a certain amount of information through statistical analysis. This makes chosen-ciphertext attack easy by analyzing the statistical characteristics of ciphertext images. Therefore, it is necessary to make the distribution of histogram even in a good cryptography.

Figure 4 shows the histograms before and after encryption for the "Lena", "Fabio", and "Pepper". We can find that the histogram for each of plaintext images is uneven before encryption, and the corresponding histogram of ciphertext image becomes even after encryption by comparing its histogram. Therefore, the proposed algorithm can make statistical analysis difficult.

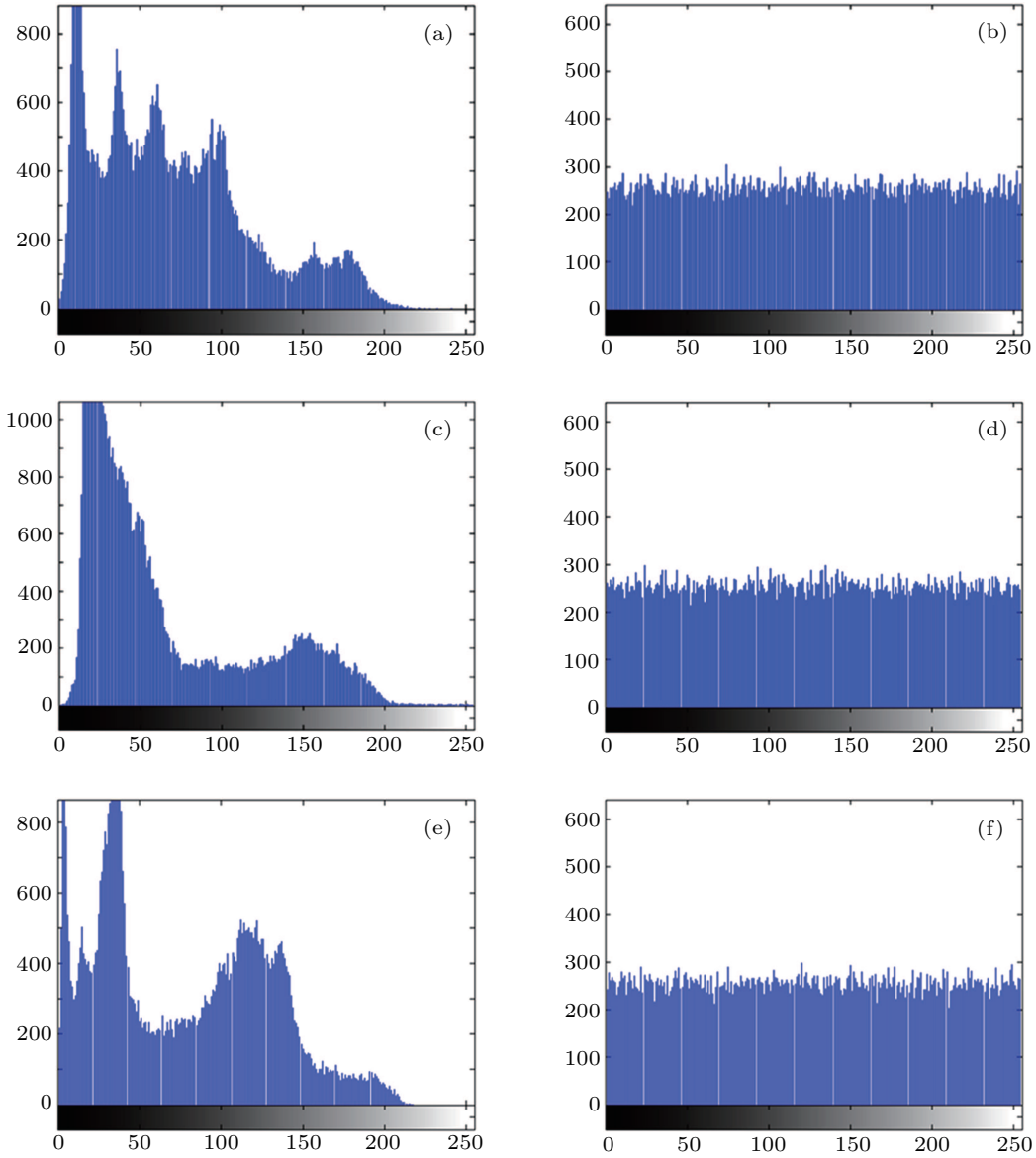


Fig. 4. Histogram analysis. (a) histogram of plaintext Lena, (b) histogram of ciphertext image Lena, (c) histogram of plaintext Fabio, (d) histogram of ciphertext image Fabio, (e) histogram of plaintext Pepper, and (f) histogram of ciphertext image Pepper.

4.3.3. Correlation between adjacent pixels

There is a high correlation between adjacent pixels in plaintext images, which is vulnerable to statistical attack. Therefore, it is necessary to reduce the correlation between adjacent pixels. We utilize Eqs. (16) and (17) to calculate the correlation between adjacent pixels of plaintext image and ciphertext image.[30] In this paper, randomly selected are 3000 pairs of adjacent pixels in the plaintext image "Lena" and its corresponding ciphertext image from each direction. The correlation between adjacent pixels is tested from horizontal, vertical and diagonal directions and the results are shown in Fig. 5.

$$r_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{16}$$

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

$$D(x) = \frac{1}{N}\sum_{i+1}^{N}(x_i - E(x))^2, \quad E(x) = \frac{1}{N}\sum_{i=1}^{N}x_i. \tag{17}$$

The correlation coefficients of plaintext image and ciphertext image "Lena" are listed in Table 5. We can find that the correlation between adjacent pixels in plaintext images is high, and it becomes low in ciphertext images after being encrypted from Table 5. And here we also give data of "Lena" in Refs. [31]–[34]. Thus, the proposed algorithm can effectively resist statistical attack.

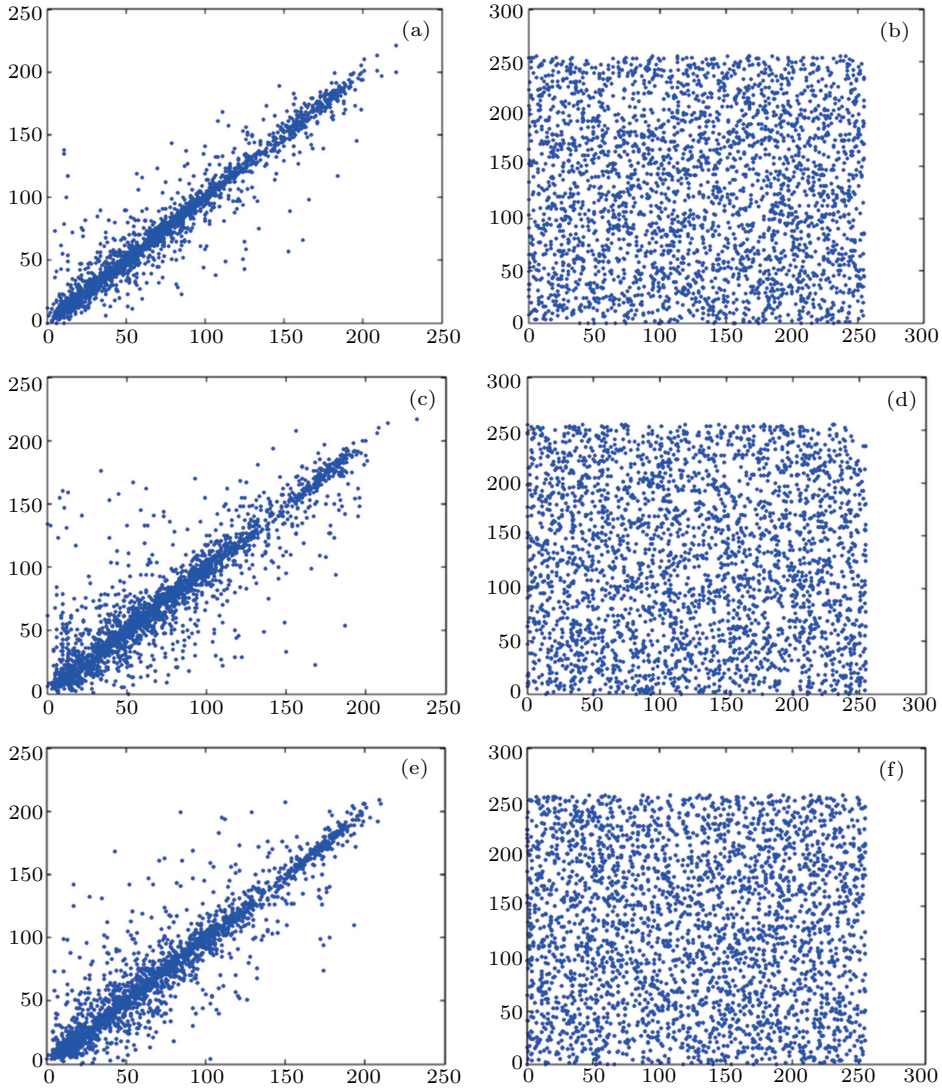


Fig. 5. Correlation between adjacent pixels of Lena. (a) Horizontal correlation in plaintext, (b) horizontal correlation in ciphertext, (c) vertical correlation in plaintext, (d) vertical correlation in ciphertext, (e) diagonal correlation in plaintext, and (f) diagonal correlation in ciphertext.

Table 5. Correlation between adjacent pixels of plaintext and ciphertext.

Test image	Horizontal	Vertical	Diagonal
Plaintext Lena	0.9721	0.9739	0.9705
Ciphertext Lena	−0.0031	0.0084	−0.0007
Ref. [31]	−0.0029	−0.0017	0.0004
Ref. [32]	0.0039	−0.0314	0.0158
Ref. [33]	−0.0020	0.0008	−0.0029
Ref. [34]	−0.475	0.196	0.135

4.3.4. Information entropy

Information entropy reflects the randomness of information, and its value can be calculated by Eq. (18).[35] The more it gets close to 8, the less possible for the algorithm to divulge information it is. The information entropies of plaintext and ciphertext images of "Lena" in different channels are listed in Table 6, and compared with the information entropy of ciphertext image in Refs. [31]–[34]

$$H(s) = \sum_{i=0}^{2^L-1} p(s_i) \log_2 \frac{1}{p(s_i)}, \tag{18}$$

where $p(s_i)$ denotes the probability of symbol s_i. The information entropy of the ciphertext image should be close to 8 after being encrypted. We can conclude that it is difficult to divulge the information after being encrypted by utilizing our proposed algorithm from Table 6. Thus, this algorithm has good performances in information entropy and it can resist the statistical attack.

Table 6. Information entropy of ciphertext images.

Test image	Information entropy of cipher
Lena (256×256)	7.9971
Fabio (256×256)	7.9975
Pepper (256×256)	7.9974
Ref. [31]	7.9971
Ref. [32]	7.9890
Ref. [33]	7.9999
Ref. [34]	7.9998

4.3.5. Sensitivity analysis to plaintext

For the algorithm in this paper, when the plaintext changes, λ will change even when the *Key* is the same and

the ciphertext will be completely different. So it is difficult to choose a plaintext attack on this encryption system. The two important features of differential attack are the number of pixels change rate (NPCR) and unified average changing intensity (UACI). Their values can be calculated from Eqs. (19) and (20) as follows:[36]

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100, \tag{19}$$

$$UACI = \frac{1}{M \times N}\left[\sum_{i,j} \frac{|c_1(i,j) - c_2(i,j)|}{255}\right] \times 100. \tag{20}$$

Here M and N indicate the width and height of the image, c_1 is the original plaintext encrypted image, and c_2 is the ciphertext image after changing a pixel value. If $c_1(i,j) \neq c_2(i,j)$, then $D(i,j) = 1$, else $D(i,j) = 0$. We make a 1-bit change for the plaintext image "Lena". The experimental results are shown in Table 7.

Table 7. NPCR and UACI values for ciphertext images.

Test image	NPCR/%	UACI/%
Lena (256×256)	99.6016	33.4735
Fabio (256×256)	99.6102	33.4811
Pepper (256×256)	99.6057	33.4375
Ref. [31]	99.5986	33.4561
Ref. [32]	99.6185	28.7344
Ref. [33]	99.9900	33.4400
Ref. [34]	99.6200	33.4600

In general, the values of NPCR and UACI must approach to 99.6093% and 33.4635% respectively. As seen in Table 7, the NPCR and UACI values represent the superior performance of our proposed algorithm in terms of diffusion.

4.3.6. Robustness analysis

Robustness is an important feature to test the capacity of resisting disturbance of a cryptography.[37] We utilize noise attack and occlusion attack to test the robustness of our proposed algorithm.

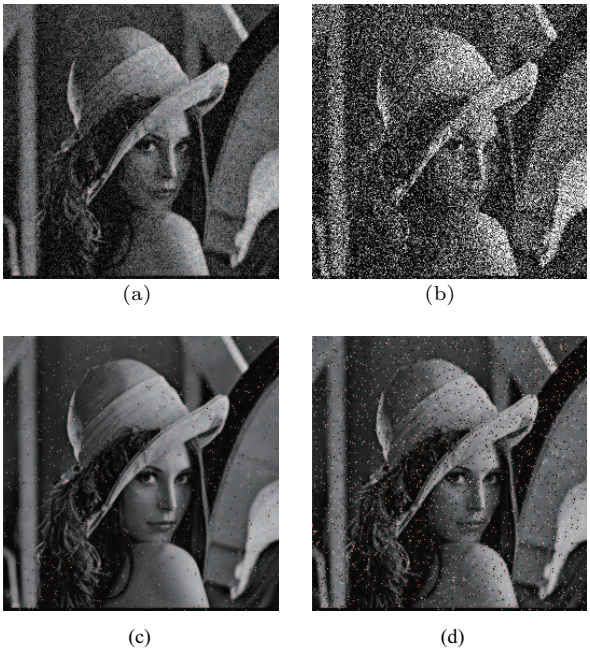


Fig. 6. Experimental results for a noise attack. Gaussian noise decoding results with variances of (a) 0.01 and (b) 0.1. Salt and Pepper noise decoding results with intensity of (c) 0.01 and (d) 0.05.

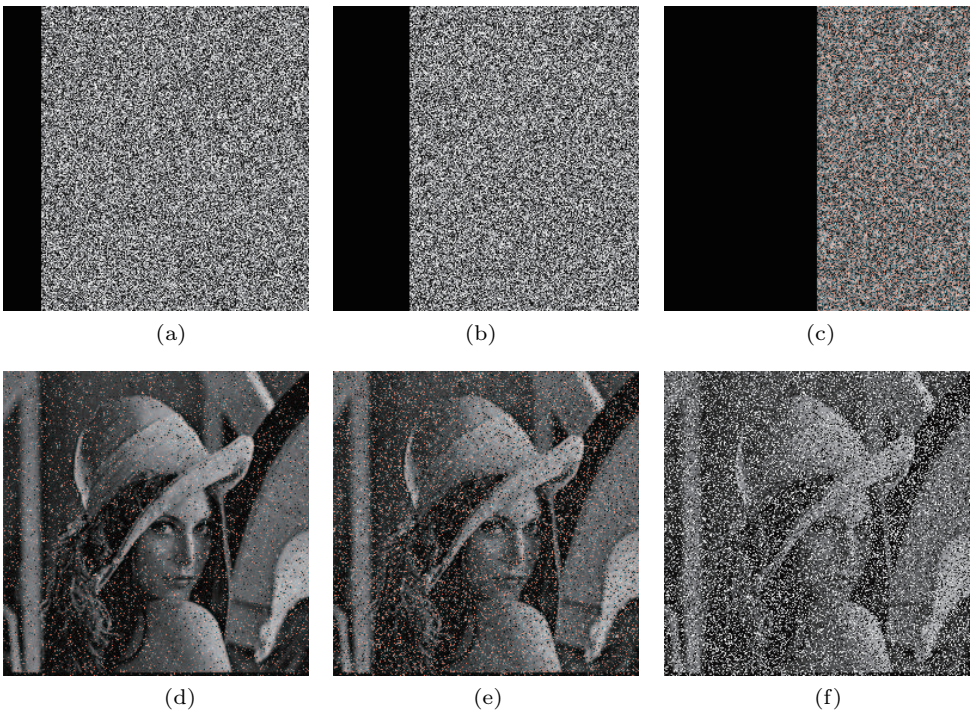


Fig. 7. Experimental results for an occlusion attack. Encryption images with (a) 1/8, (b) 1/4, and (c) 1/2 occlusion; panels (d), (e), and (f) are the corresponding decryption images.

In the process of actual communication, one of the most important problems is noise interference. The error propagation phenomenon implies that errors in the encryption image will lead to errors in the decryption image. The common noise types are Gauss noise, salt and Pepper noise, etc. We add Gaussian noise with different variances and salt and Pepper noise with different intensities to the plaintext images, and utilize the same key to encrypt and decrypt. Figure 6 shows decryption images after adding Gauss noise with variances of 0.01 and 0.1 and salt and Pepper noise with intensity 0.01, 0.05. We can see that under a limited amount of noise, the decryption image can be identified. Thus, our proposed algorithm can resist the noise attack.

Occlusion attack is another kind of robustness test, and we decrypt the images which are the encryption images with 1/8, 1/4, and 1/2 occlusion. The quality of decryption images decreases as the occlusion size increases. Figure 7 shows the encryption images with different sizes of occlusions and corresponding decryption images. The main information of plaintext images can be identified from the decryption images obtained from experimental results. Therefore, the proposed algorithm has high robustness and it can also resist the occlusion attack.

5. Conclusions

We proposed a new chaotic image encryption algorithm based on Fisher–Yatess scrambling and DNA coding in this paper. The algorithm includes the following three major improvements. The Fisher–Yatess method that we use in the scrambling process is efficient and safe. It solves the shortcomings of some traditional methods which are periodic. The methods such as Arnold transform and Baker transform are periodic. We then simplify the DNA method. It becomes more efficient, which is what we are pursuing. Finally, we add plaintext statistics for pixel-level diffusion. This makes our algorithm perform better against plaintext attacks. The algorithm uses high-efficiency low-dimensional chaos, thus avoiding the generation of long chaotic sequences. Our experimental results demonstrate that the algorithm can resist common attacks, and has an advantage in encryption time over the existing methods.

References

[1] Wang L D, Xiao L Q, Li Q D and Shu K 2018 *Acta Phys. Sin.* **67** 090502 (in Chinese)
[2] Zhou N R, Pan S M and Cheng S 2016 *Opt. Laser Technol.* **82** 121
[3] Akram B, Ahmed A and Adrian V D 2017 *Opt. Lasers Eng.* **88** 37
[4] Kannammal A and Rani S S 2014 *Int. J. Im. Syst. Technol.* **24** 111
[5] Potlapally N R, Ravi S, Raghunathan A and Jha N K 2006 *IEEE Trans. Mob. Comput.* **5** 128
[6] Tan J T, Luo Y R, Zhou Z and Hai W H 2016 *Chin. Phys. Lett.* **33** 070302
[7] Liu L F and Miao S X 2017 *Mult. Tools & Appl.* **76** 16511
[8] Chai X L, Gan Z H and Yuan K 2017 *Chin. Phys. B* **26** 020504
[9] Liu L F, Miao S X and Hu H P 2016 *Sign. Proc.* **10** 1096
[10] Liu Z Y, Xia T C and Wang J B 2018 *Chin. Phys. B* **27** 030502
[11] Li C B, Wesley J T, Julien C S 2017 *Chin. Phys. B* **26** 120501
[12] Muruga B, Nanjappa G and Ammasai G 2016 *Iet Comput. Vis.* **10** 593
[13] Tang Y X, Abdul J M K, Karthikeyan R, Viet T P, Sajad J and Tian Y 2018 *Chin. Phys. B* **27** 040502
[14] Liu S, Guo C L and Sheridan J T 2014 *Opt. Laser Technol.* **57** 327
[15] Jiang Q L, Zeng W, Ou W H and Xu R H 2016 8*th International Conference on Wireless Communications & Signal Processing,* October 13–15, 2016, Yangzhou, China, p. 1
[16] Coppersmith D 1994 *IBM J. Res. Dev.* **38** 243
[17] Liu H J and Wang X Y 2011 *Opt. Commun.* **284** 3895
[18] Wang L, Song H and Liu P 2016 *Opt. Lasers Eng.* **77** 118
[19] Erlich Y and Zielinski D 2017 *Science* **355** 950
[20] Leier A, Richter C, Banzhaf W and Rauhe H 2000 *Biosystems* **57** 13
[21] Shimanovsky B, Feng J and Potkonjak M 2008 *Lect. Notes Comput. Sci.* **2578** 373
[22] Wang J S, Long F and Ou W H 2018 *International Conference on Security, Pattern Analysis, and Cybernetics*, December 15–17, 2017, Shenzhen, China, p. 730
[23] Gehani A and Reif J 1999 *Biosystems* **52** 197
[24] Chen J 2003 *Proceedings of the 2003 International Symposium on Circuits and Systems*, May 25–28, 2003, Bangkok, Thailand, p. 822
[25] Tanaka K, Okamoto A and Saito I 2005 *Biosystems* **81** 25
[26] Mousa H, Moustafa K, Abdel W and Hadhoud M 2008 *Int. Arab. J. Inf. Technol.* **8** 147
[27] Liu Y, Zang Q and Wie X 2014 *Opt. Laser Technol.* **60** 111
[28] Siavash B S, Mehran M K and Arash R M 2014 *IEEE Trans. Comp. Aided . Int. Circ. Syst.* **33** 1105
[29] Alvarez G and Li S J 2006 *Int. J. Bifur. Chaos* **16** 2129
[30] Rhouma R, Meherzi S and Belghith S 2009 *Chaos, Solitons, and Fractals* **40** 309
[31] Wang X Y, Zhu X Q and Zhang Y Q 2018 *IEEE Access* **6** 23733
[32] Zhang X C, Zhou Z and Niu Y 2018 *IEEE Phot. J.* **10**
[33] Zhu C X and Sun K H 2018 *IEEE Access* **6** 18759
[34] Fu X Q, Liu B C, Xie Y Y, Li W and Liu Y 2018 *IEEE Phot. J.* **10**
[35] Awad A 2010 *Etri. J.* **32** 774
[36] Wu Y 2011 *J. Sel. Areas Tel.* **2** 31
[37] Zhou N R, Wang Y X, Gong L H, Chen X B and Yang Y X 2012 *Opt. Laser Technol.* **44** 2270