IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 18, 2020, accepted June 29, 2020, date of publication July 2, 2020, date of current version July 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3006513

An Efficient and Secure Image Encryption Algorithm Based on Non- Adjacent Coupled Maps

HAO ZHANG[1], (Member, IEEE), XIAOQING WANG[1], HONGWEI XIE[2], CHUNPENG WANG[3], (Member, IEEE), AND XINGYUAN WANG[4]

[1]College of Information and Computer, Taiyuan University of Technology, Taiyuan 030024, China
[2]College of Software, Taiyuan University of Technology, Taiyuan 030024, China
[3]School of Cyber Security, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China
[4]School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China

Corresponding author: Hao Zhang (zhangh545@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702356, Grant 61802212, Grant 61672124, and Grant 61503375, in part by the Natural Science Foundation of Shanxi Province under Grant 201801D121143, and in part by the Password Theory Project of the 13th Five-Year Plan National Cryptography Development Fund under Grant MMJJ20170203.

ABSTRACT In this paper, an image encryption algorithm based on non-adjacent coupled map lattices is designed. Compared with coupled map lattices, non-adjacent coupled map lattices have few periodic windows in bifurcation, and larger parameter range in chaos dynamics. Combined with the proposed chaotic system, a new alternative encryption structure between permutation, diffusion and substitution is proposed to improve the security and encryption efficiency of the algorithm. In the diffusion process, multiplication operation in GF (257) and addition operation in GF (2^8) based on look-up table are adopted, which improves the encryption efficiency and enhances the permutation effect by choosing pixel value in Latin Square during the diffusion process. Moreover, based on the good cryptographic characteristics of the s-box, a dynamic s-box related to plaintext is constructed to substitute plaintext pixels. Experimental results and security analysis show that the proposed encryption scheme can resist all types of typical attacks and has good encryption effect.

INDEX TERMS SHA-256, non adjacent coupled map lattices, dynastic s-box, Latin square, 3D Arnold GF (257), GF (2^8).

I. INTRODUCTION

With the rapid development of Internet communication technology, more and more images are transmitted through the Internet. Therefore, image storage and transmission become more and more important. Because the image data has the characteristics of high redundancy and strong correlation between adjacent pixels. Therefore, the traditional text encryption method (DES, AES, RSA) is not suitable for efficient, real-time encryption of images. Therefore, special encryption schemes are proposed by combining with different technologies, such as chaotic system [1]–[4], cellular automata [5], DNA operation [6], CS [7], [8].

As a pseudo-random generator, chaos system is widely used in image encryption because of its sensitivity to initial values and parameters, pseudo randomness, aperiodicity and other characteristics [1], [2], [9]–[11]. However, due to the limited computing accuracy of the computer [12], any chaos system has periodicity, which reduces the security of the cryptosystem. Therefore, in order to improve the security of cryptosystem, many scholars use spatiotemporal chaos system to generate key sequence. Compared with single chaotic mapping, spatiotemporal chaotic system has a wider parameter range and chaotic properties. As a typical spatiotemporal chaotic system, coupled map lattices (CML) has been widely used in image encryption [6], [13]. However, some lattices do not have chaotic behavior since CML coupling adjacent lattices [14], which is proved to be unsafe. In order to solve these problems, some scholars have proposed non-adjacent coupled mapping lattice [15]–[17].

Since Fridrich first proposed the scheme of image encryption based on chaos in 1998 [18], most image encryption methods adopt permutation-diffusion encryption model, which has been proved to have good performance [19]–[23]. since permutation and diffusion operations are carried out separately and independently, and in order to achieve good

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Asikuzzaman.

diffusion effect, permutation and diffusion operations for the same pixel value are usually performed in multiple rounds, which will undoubtedly increase the complexity of the algorithm and reduce the encryption efficiency. Such as, Alireza *et al.* propose an image encryption method based on chaos system and AES algorithm [19], which improves the security through multiple rounds of permutation and diffusion encryption. In order to extend the effect of changing one pixel of the plaintext image to all pixels, a second round of exclusive or operation is added in ref. [20]. In ref. [21], the diffusion layer and the substitution layer are repeatedly executed three times to improve the security of the algorithm. Chen *et al.* proposed an improved image encryption algorithm based on chaotic mapping [22]. The same pixel performs two encryption operations, namely row encryption and column encryption. Dua, M. *et al.* proposed a color image encryption method based on bit level permutation and alternating logistic mapping [23]. The scrambling and diffusion of the three channels are carried out separately. Based on the above problems, some scholars improve the plaintext sensitivity by combining the hash function with the initial value and parameters of the chaotic system [24]–[26], so that the permutation and diffusion operations can only be performed once on the overall image, and achieve efficient and secure encryption.

As a non-linear component, S-box is widely used in AES and DES algorithm to enhance the effect of confusion. This encryption technology is widely used in image encryption because of its simple implementation and easy embedding [27]–[29]. Such as, Ben Farah *et al.* [27] proposed an image encryption scheme based on mixed chaotic map and optimized s-box. This encryption scheme uses Jaya algorithm to optimize the s-box generated from chaotic sequence. The nonlinear criterion test and output bit independent criterion test of the optimized s-box have better performance. In ref. [28], the s-box is constructed by using the spatial position of Hilbert curve, and the value of the plaintext pixel is substituted by the S-box.

Based on the above analysis, an efficient and secure image encryption algorithm based on non- adjacent coupled map is proposed in this paper. Firstly, the CML is combined with 3D Arnold transformation, and a non-adjacent coupled map model is proposed to improve the chaotic range and security of the system. Secondly, the structure of permutation, diffusion and substitution alternately execution is adopted and each pixel value is encrypted only once, which reduces the complexity of the algorithm and improves the security. Finally, in the process of constructing the s-box, the irreducible polynomials are chosen dynamically by the hash values related to the plaintext, and then the plaintext pixel values are replaced by dynamic s-box.

The rest of this paper is organized as follows. Section 2 introduces the no-adjacent coupled map lattices and Latin squares. Section 3 includes the proposed encryption algorithm. Section 4 analyzes the security performance of proposed scheme. A conclusion is drawn in Section 5

II. RELATED WORK

A. NON-ADJACENT COUPLED MAP LATTICES (NACML)

1) COUPLED MAP LATTICES

The CML is a spatiotemporal chaos system, which has a larger key space. The finite computing precision makes any chaotic system periodic, but the period of CML system is long enough to ensure the security of key information, which is defined in Eq. (1).

$$\begin{cases} x_{n+1}(i) = (1-\varepsilon)f(x_n(i)) + \dfrac{\varepsilon}{2}[\, f(x_n(i-1)) \\ \qquad\qquad\qquad\qquad + f(x_n(i+1))] \\ f(x) = \mu x(1-x) \end{cases} \tag{1}$$

where i=1,2,…,L. is the ith lattice, i-1, i+1, which are two adjacent lattices. N is the time index. $\varepsilon \in (0, 1)$ is the coupling parameter and $\mu \in (3.57, 4]$ is the control parameter. For the traditional CML, each lattice is only coupled into chaos state by two adjacent lattices. The diffusion effect is weak, and the energy distribution among the lattice is unevenly. Therefore, for this reason, we propose a non adjacent coupled mapping model. 3D Arnold cat mapping is used to establish the relationship between i-1, i, and i + 1.

2) 3D ARNOLD MAP

The 3D Arnold map is a discrete high-dimensional Arnold map. Compared with 2D Arnold map, it has more key space. It is described as follow:

$$\begin{bmatrix} x_{n+1\_} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = A \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \left(mod\, 2^L \right). \tag{2}$$

where A is a 3×3 matrix, providing chaotic behavior. Specifically,

$$A = \begin{bmatrix} 1 & a_x + a_y b_z & a_x a_z + a_y(1 + a_z b_z) \\ b_x & 1 + a_x b_x + b_x a_y b_z & (1 + a_x b_x)\, a_z + b_x a_y (1 + a_z b_z) \\ b_y & (1 + a_y b_y) b_z & (1 + a_y b_z)(1 + a_z b_z) \end{bmatrix},$$

the inverse matrix is:

$$A^{-1} = \begin{bmatrix} 1 + a_y b_y + a_x b_x (1 + a_y b_y) & -a_x - a_x a_y b_y & -a_y \\ -b_x - b_x a_z b_z + b_y a_z (1 + a_x b_x) & 1 + a_z b_z - b_y a_x a_z & -a_z \\ b_x b_z - b_y - b_y a_x b_x & -b_z + b_y a_x & 1 \end{bmatrix}.$$

The $a_x, a_y, a_z, b_x, b_y, b_z$ is positive integers, and they are all set to 1. In this paper, 3D Arnold map is used to couple adjacent lattice i-1, i, i + 1 to eliminate the energy diffusion inhomogeneity caused by local coupling, as shown in Eq. (3).

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = A \begin{bmatrix} (i-1) \\ i \\ (i+1) \end{bmatrix} (mod\ i). \tag{3}$$

Fig. 1 shows the NACML of 3D Arnold map coupled with CML with L=500, $\varepsilon = 0.3$, $\mu = 3.99$. It can be seen that the proposed system has complex spatiotemporal chaos behavior, and the generated pseudo-random sequences are evenly distributed in the whole space. Therefore, the system has very good chaotic properties.

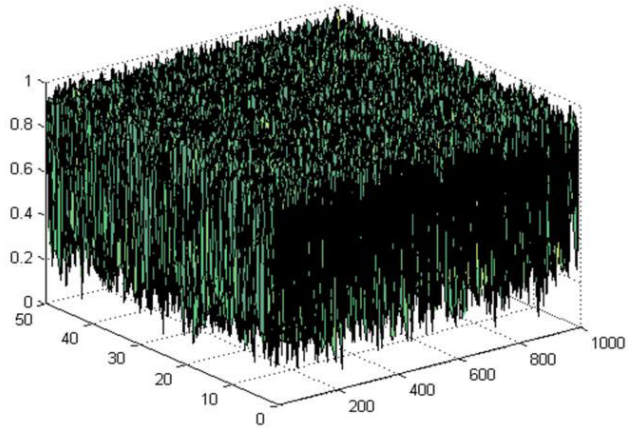

FIGURE 1. Spatiotemporal chaos.

B. NIST SP800-22 TEST OF CHAOTIC SEQUENCES

In order to evaluate the randomness of chaotic sequences, NIST SP800-22 test, which include 15 groups test methods, is used to test the randomness of bit sequences. Supposed the value P is greater than 0.01, then the tested sequence is random. In this paper, we set the initial values of six sets of sequences as: $x_1(1) = 0.8147$, $x_2(1) = 0.9058$, $x_3(1) = 0.1270$, $x_4(1) = 0.9134$, $x_5(1) = 0.6324$, $x_6(1) = 0.0975$, parameters $\varepsilon_0 = 0.1365, \mu_0 = 3.9938$ and iterate the chaotic system 10^6 times, and then choose a set of chaotic sequences to convert into 0-1 bit sequences to test the randomness. As shown in Table 1, all test results of P value are greater than 0.01, so the chaotic sequence has randomness.

C. LATIN SQUARES

N-order Latin square is a square matrix of $n \times n$ composed of N element sets $S = 0, 1, \ldots, n-1\}$, in which each element occurs once in each row and column. In this paper, we set $n = M = N$, and use $L(i, j)$ to represent the elements of row i and column j in Latin square L. In order to construct a matrix LS that can traverse all pixels of image, we construct a Latin square L at first, and subsequently, we use the Latin square L to construct the matrix LS for the image scrambling phase of encryption algorithm. The specific construction process is as follows:

First of all, we construct Latin square L with size $n = 2 \times m$, where n is a positive even number greater than 2. The first element of matrix L is $0, 1, 2m-1, 2, 2m-2, 3, 2m-3, \ldots, m-1, m+1, m$. Following, the other elements in each row are obtained by adding 1 ($MODn$) to the element in the previous row. At last, the matrix LS is constituted by element pairs of adjacent row elements in matrix L. For example

$$L_{(6\times6)} = \begin{bmatrix} 0 & 1 & 5 & 2 & 4 & 3 \\ 1 & 2 & 0 & 3 & 5 & 4 \\ 2 & 3 & 1 & 4 & 0 & 5 \\ 3 & 4 & 2 & 5 & 1 & 0 \\ 4 & 5 & 3 & 0 & 2 & 1 \\ 5 & 0 & 4 & 1 & 3 & 2 \end{bmatrix}$$

TABLE 1. NIST SP800-22 test results of binary sequences generated using NACML.

Test	$\mathbf{P-value} \geq \mathbf{0.01}$	Pass or not
Frequency (Monobit) test	0.2485	Pass
Frequency test	0.6897	Pass
Runs test	0.7923	Pass
Longest-run-of-ones in a block	0.1694	Pass
Binary matrix rank test	0.1562	Pass
Discrete Fourier transform test	0.2341	Pass
Non-overlapping template matching test	0.4103	Pass
Overlapping template matching test	0.4747	Pass
Maurer's universal statistical test	0.5245	Pass
Liner complexity test	0.5219	Pass
Serial test (p-value1)	0.4944	Pass
Serial test (p-value2)	0.7431	Pass
Approximate entropy test	0.3008	Pass
Cumulative sums test (Forward)	0.6110	Pass
(Backward)	0.9185	Pass
Random excursions test (x=-4)	0.0340	Pass
Random excursions test (x=-3)	0.8011	Pass
Random excursions test (x=-2)	0.9282	Pass
Random excursions test (x=-1)	0.6973	Pass
Random excursions test (x=1)	0.6648	Pass
Random excursions test (x=2)	0.9317	Pass
Random excursions test (x=3)	0.5517	Pass
Random excursions test (x=4)	0.2441	Pass
Random excursions variant test (x=-9)	0.0976	Pass
Random excursions variant test (x=-8)	0.1139	Pass
Random excursions variant test (x=-7)	0.1289	Pass
Random excursions variant test (x=-6)	0.1745	Pass
Random excursions variant test (x=-5)	0.2313	Pass
Random excursions variant test (x=-4)	0.3916	Pass
Random excursions variant test (x=-3)	0.8745	Pass
Random excursions variant test (x=-2)	0.7597	Pass
Random excursions variant test (x=-1)	0.4984	Pass
Random excursions variant test (x=1)	0.4984	Pass
Random excursions variant test (x=2)	0.6221	Pass
Random excursions variant test (x=3)	0.6169	Pass
Random excursions variant test (x=4)	0.6563	Pass
Random excursions variant test (x=5)	0.5962	Pass
Random excursions variant test (x=6)	0.6961	Pass
Random excursions variant test (x=7)	0.5956	Pass
Random excursions variant test (x=8)	0.2907	Pass
Random excursions variant test (x=9)	0.2304	Pass

$$\Longrightarrow LS = \begin{bmatrix} (0,1) & (1,5) & (5,2) & (2,4) & (4,3) & (3,3) \\ (1,2) & (2,0) & (0,3) & (3,5) & (5,4) & (4,4) \\ (2,3) & (3,1) & (1,4) & (4,0) & (0,5) & (5,5) \\ (3,4) & (4,2) & (2,5) & (5,1) & (1,0) & (0,0) \\ (4,5) & (5,3) & (3,0) & (0,2) & (2,1) & (1,1) \\ (5,0) & (0,4) & (4,1) & (1,3) & (3,2) & (2,2) \end{bmatrix}$$

D. MULTIPLICATION AND ADDITION IN GALOIS FIELD

Finite field plays an important role in cryptography. The order of finite field, that is, the number of elements must be a power of prime number. The finite field of order p^n is generally recorded as $GF(p^n)$, GF represents Galois field, p is a prime number, n is a positive integer. In order to speed up the operation in Galois Fine, a multiplication look-up table and an addition look-up table are constructed before encryption. The addition table AT which is constructed by using the default irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$ and addition operation on $GF(2^8)$ is shown

Algorithm 1 The Construction of Addition Table at and Multiplication Table MT

1: **for** i←0 to 2^8-1 **do**
2: **for** j←1 to 8 **do**
3: a(j)←mod ($\left\lfloor \frac{i}{2^{(8-j)}} \right\rfloor$, 2)
4: **end for**
5: **for** j←0 to 2^8-1 **do**
6: **for** k←1 to 8 **do**
7: b(k)←mod ($\left\lfloor \frac{j}{2^{(8-k)}} \right\rfloor$, 2)
8: **end for**
9: t←mod(a+b,2);
10: AT(i + 1, j + 1) ←$sum(t.\times 2^{(-7to1)})$);
11: **end for**
12: **end for**
13: MT←mod (transpose(0 to 256) ×(0 to 256),257);

in algorithm 1. However, in order to prevent data loss during decryption, multiplication operation defined *onGF*(257) field is used, the constructed multiplication table MT on $GF(257)$ and addition table AT on $GF(2^8)$ are shown in Algorithm 1.

III. THE PROPOSED ENCRYPTION SCHEME

A. INITIALIZATIONS OF SYSTEM PARAMETERS AND INITIAL VALUES

In this paper, in order to improve the plaintext sensitivity of the algorithm, The SHA-256 hash function of plain image is used to generate 256-bit external key, and then use the external key to initialize the system parameters and initial values. The 256-bit external key K is divided into 32 floating-point numbers with 8 bits as a block $k_1, k_2, \ldots, k_{32}$. The initialization system parameters and initial values can be calculated by Eq. (4).

$$\begin{cases} t_i' = t_i + \dfrac{(k_i \oplus k_{i+8} \oplus k_{i+16}) + t\,(i)}{(k+t)\times 10^3} \\ t = \dfrac{sum\left(mod\left(t_i \times 10^{14}\right), 256\right)}{8}, \end{cases} \tag{4}$$

where $k = k_1 \oplus k_2 \oplus \cdots k_{32}$. $t_i (i = 1, 2, .., 8)$ denotes the given parameter ε_0, μ_0 and initial value $x_1(1), x_2(1), x_3(1), x_4(1), x_5(1), x_6(1)$ respectively. t_i' represents the system parameter and initial value initialized by the given parameter, initial value and hash value K.

B. THE CONSTRUCTION OF S-BOX

As a nonlinear component, S-box is widely used in cryptography. The traditional encryption algorithm usually uses fixed irreducible polynomials to construct the S-box. In this paper, a dynamic method is used to construct the S-box, namely, SHA-256 hash value is used to select the polynomials of the S-box, so as to dynamically generate the S-box related to plaintext, all the primitive irreducible polynomials with degree 8 are shown in Table 2. The construction flow of the *Sbox_r* of R-channel is shown in Fig. 2(a).

TABLE 2. Irreducible polynomials of degree 8.

Binary	Binary	Binary	Binary	Binary
100011011	101001101	101110111	110100011	111010111
100011101	101011111	101111011	110101001	111011101
100101011	101100011	110000111	110110001	111100111
100101101	101100101	110001011	110111101	111110011
100111001	101101001	110001101	111000011	111110101
100111111	101110001	110011111	111001111	111111001

Similarly, the *Sbox_g*, *Sbox_b* is generated using *poly_g*, *poly_b* based on flow chart of Fig. 2(a), respectively. Fig. 2(b) is the construction flow of the inverse S-box *Inv_sbox_r* of R-channel, which is used in decryption. Similarly, the *Inv_Sbox_g*, *Inv_Sbox_b* is generated according to flow chart of Fig. 2(b) by using *poly_g*, *poly_b*, respectively. The "inversion in $GF(2^8)$" represents the inverse operation on $GF(2^8)$ on Fig. 2(b), which is calculated by the Extended Euclidean algorithm [30]. In the construction process, the construction of S-box and inverse S-box is based on constant binary key matrix, so that the construction of inverse S-box is independent of S-box, thus reducing the waste of storage space caused by saving S-box to construct inverse S-box.

Here, we use the hash values $k_{25}, k_{26}, \ldots, k_{32}$ of SHA-256 to calculate the irreducible polynomials of three channels, *poly_r*, *poly_g*, *poly_b*, as follows in Eq. (5), Eq.(6).

$$\begin{cases} \mathbf{T_r} = \mathbf{bin2dec(stract(dec2bin(k_{25}), dec2bin(k_{26}), dec2bin(k_{27})))} \\ \mathbf{T_g} = \mathbf{bin2dec(stract(dec2bin(k_{28}), dec2bin(k_{29}), dec2bin(k_{30})))} \\ \mathbf{T_b} = \mathbf{bin2dec(stract(dec2bin(k_{31}), dec2bin(k_{32}), dec2bin(k_{33}))),} \end{cases} \tag{5}$$

$$\begin{cases} poly\_r = mod\,(T_r, 30) + 1 \\ poly\_g = mod\left(T_g, 30\right) + 1. \\ poly\_b = mod\,(T_b, 30) + 1 \end{cases} \tag{6}$$

where function *stract* is to concatenate 0-1 bit.

C. THE ENCRYPTION ALGORITHM

In this section, the proposed algorithm uses the structure of alternating between scrambling, diffusion and substitution to improve the effectiveness of the algorithm. Simultaneously, SHA-256 hash function is used to initialize the key and construct the S-box to improve the plain sensitivity, so as to enhance the ability of the algorithm to resist chosen plaintext attacks. Fig. 3 shows the flow chart of encryption algorithm. The detailed steps are as follows:

1) QUANTIZATION OF PSEUDO-RANDOM SEQUENCE

Before encryption, the key stream $(S_1, S_2, S_3, S_4, S_5, S_6)$ generated by chaotic system is quantized K_{diff} and K_{comp} and then

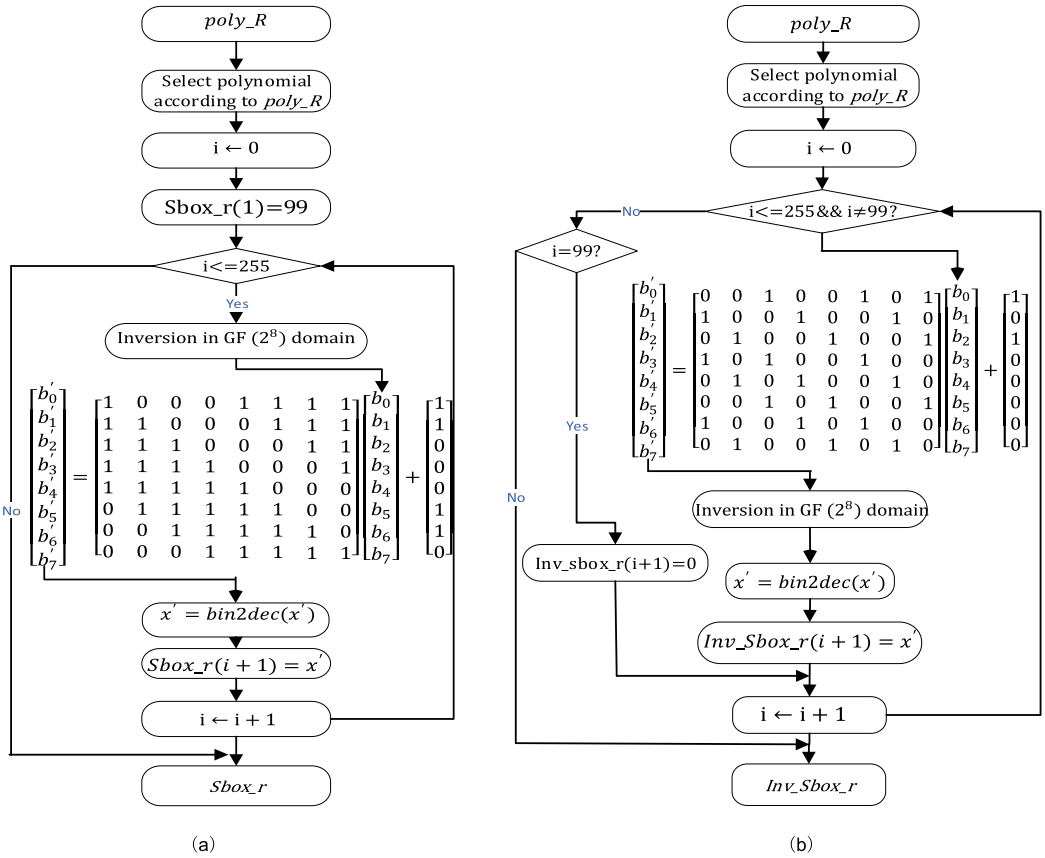

FIGURE 2. The construction of S-box and inverse S-box in R-channel: (a) S-box, (b) Inverse S-box.

used for diffusion and permutation. The detailed process is shown in Algorithm 2.

2) ENCRYPTION PROCESS

In order to improve the security of the algorithm, the structure encryption of diffusion, permutation and substitution alternate transform is adopted. Each pixel is encrypted only once, so the complexity of the algorithm is reduced. The multiplication and addition operations in the finite field involved in the diffusion operation are realized by look-up the table, so as to speed up the encryption speed. The specific encryption steps are as follows.

Step 1: Initialize Latin square matrix LS=zeros (M, N); LS (1,1) = 0; LS (1,2) = 1; LS (1,3) = M-1.

Step 2: The pixel P (i, j) (i=1, j=1 to N) of the image P is encrypted as follows:

1) For the fourth to N-1 pixels in the first row, calculate the Latin square according to Eq. (9)

$$\begin{cases} LS\,(i,j+1) = LS\,(i,j-1) + 1\,(j+1)\,/2 = 0 \\ LS\,(i,j+1) = LS\,(i,j-1) - 1\,(j+1)\,/2 \neq 0 \end{cases}. \tag{9}$$

1) The first-row pixels of R, G, B component performs the following diffusion operation.

$$\begin{cases} C^R_{i,j} = P^R_{LS(i,j)+1,LS(i,j+1)+1} \times C^R_{M,N} + K_{diff}\ (\mathrm{M, N, 1}) \\ i = 1, j = 1 \\ C^R_{i,j} = P^R_{LS(i,j)+1,LS(i,j+1)+1} \times C^R_{i,j-1} + K_{diff}\ (\mathrm{i, j, 1}) \\ j \neq 1 \\ C^R_{i,j} = P^R_{LS(i,j)+1,LS(i,j)+1} \times C^R_{i-1,j} + K_{diff}\ (\mathrm{i, j, 1}) \\ j = N, \end{cases} \tag{10}$$

$$\begin{cases} C^G_{i,j} = P^G_{LS(i,j)+1,LS(i,j+1)+1} \times C^G_{M,N} + K_{diff}\ (\mathrm{M, N, 2}) \\ i = 1, j = 1 \\ C^G_{i,j} = P^G_{LS(i,j)+1,LS(i,j+1)+1} \times C^G_{i,j-1} + K_{diff}\ (\mathrm{i, j, 2}) \\ j \neq 1 \\ C^G_{i,j} = P^G_{LS(i,j)+1,LS(i,j)+1} \times C^G_{i-1,j} + K_{diff}\ (\mathrm{i, j, 2}) \\ j = N, \end{cases} \tag{11}$$

$$\begin{cases} C^B_{i,j} = P^B_{LS(i,j)+1,LS(i,j+1)+1} \times C^R_{M,N} + K_{diff}\ (\mathrm{M, N, 3}) \\ i = 1, j = 1 \\ C^B_{i,j} = P^B_{LS(i,j)+1,LS(i,j+1)+1} \times C^B_{i,j-1} + K_{diff}\ (\mathrm{i, j, 3}) \\ j \neq 1 \\ C^B_{i,j} = P^B_{LS(i,j)+1,LS(i,j)+1} \times C^B_{i-1,j} + K_{diff}\ (\mathrm{i, j, 3}) \\ j = N. \end{cases} \tag{12}$$

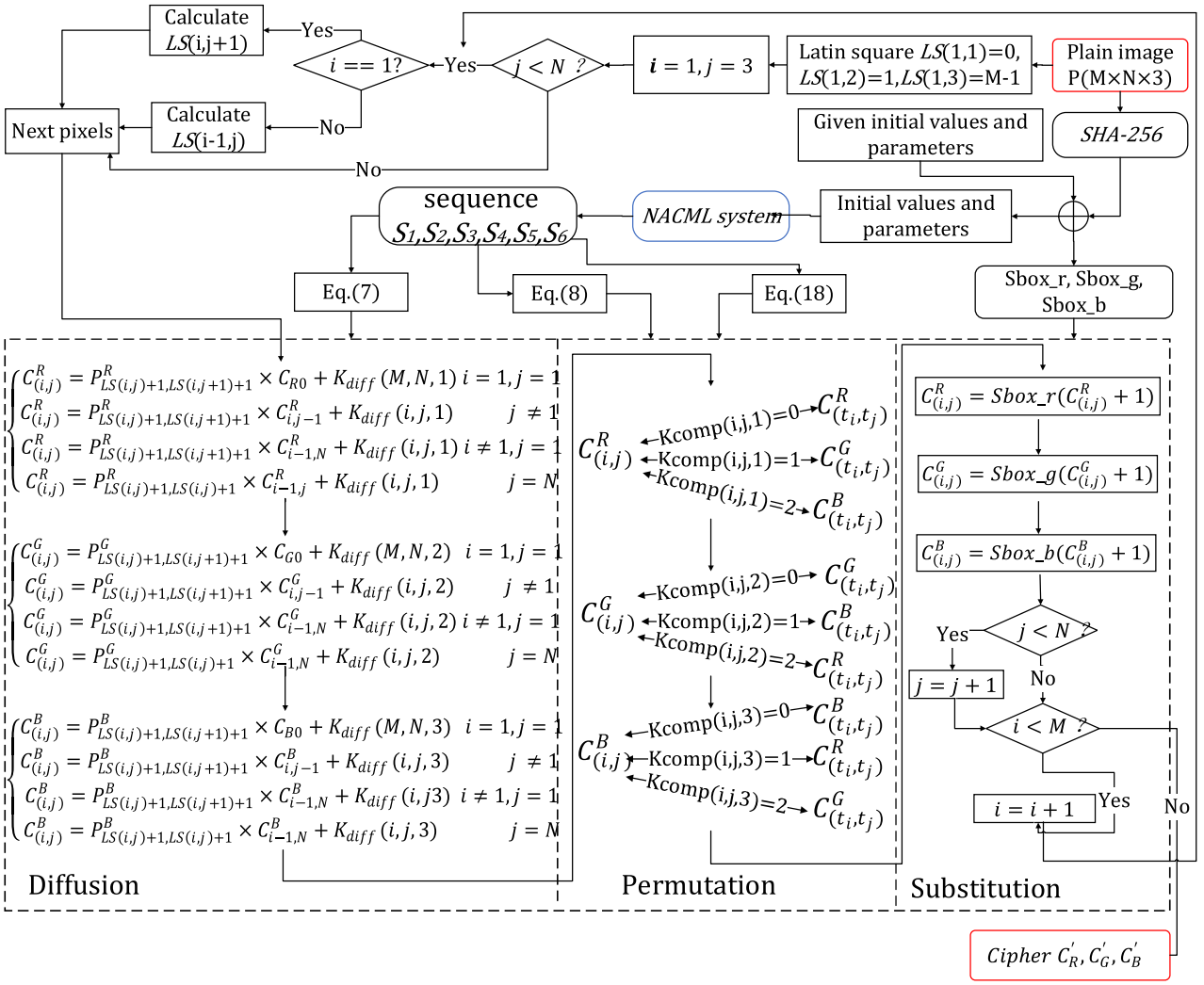

FIGURE 3. Flow chart of encryption algorithm.

where the symbols "+" and "×" denotes addition over $GF(2^8)$ and multiplication operations over $GF(257)$, respectively. Such as, "a+b" and "a×b" can be calculated by looking up tables AT, MT, respectively. Namely, $a + b = AT(a + 1, b + 1)$, $a \times b = MT(A + 2, B + 2) - 1$.

1) Connect three components of R, G and B into a three-dimensional matrix, namely, $C(:, :, 1) = C^R$, $C(:, :, 2) = C^G$, $C(:, :, 3) = C^B$.Then, perform the permutation operation as shown in Algorithm 3.
2) Perform the substitution operation by using the Eq. (13), then assign the replaced value $C(i, j, 1)$ to $C^R_{i,j}$, $C(i, j, 2)$ to $C^G_{i,j}$, $C(i, j, 3)$ to $C^B_{i,j}$.

$$\begin{cases} C(i,j,1) = Sbox\_r(C(i,j,1)+1) \\ C(i,j,2) = Sbox\_g(C(i,j,2)+1) \\ C(i,j,3) = Sbox\_b(C(i,j,3)+1). \end{cases} \tag{13}$$

Step 3: For the pixels P (i, j) (i=2 to N, j=1 to N) of the plain image P, perform the following encryption operation:

1) Calculate the Latin square according to Eq. (14).

$$\begin{cases} LS(i,j) = mod(LS(i-1,j)+1, M); \\ LS(i,j+1) = mod(LS(i-1,j+1)+1, M); j \neq N. \end{cases} \tag{14}$$

2) Perform the diffusion operation as described in Eq. (15).

$$\begin{cases} C^R_{i,j} = P^R_{LS(i,j)+1,LS(i,j+1)+1} \times C^R_{i-1,N} + K_{diff}(i, j, 1) \\ j = 1 \\ C^R_{i,j} = P^R_{LS(i,j)+1,LS(i,j+1)+1} \times C^R_{i,j-1} + K_{diff}(i, j, 1) \\ j \neq 1 \\ C^R_{i,j} = P^R_{LS(i,j)+1,LS(i,j)+1} \times C^R_{i-1,j} + K_{diff}(i, j, 1) \\ j = N, \end{cases} \tag{15}$$

Algorithm 2 The Quantization of Pseudo-Random Sequence

1: SHA-256 hash function input plain image to get 256-bit hash value K
2: Divided K into subkeys ($k_1, k_2, \ldots, k_{32}$), Turn these subkeys and t_i into parameters and initial values of NACML as shown in Eq. (2).
3: Bring $t_i^{'}$ into NACML and iterate NACML $1000+\lceil(M\times N)/2\rceil$ times. Choose the last $\lceil(M\times N)/2\rceil$ values of each group to form pseudo-random sequence $S_1, S_2, S_3, S_4, S_5, S_6$.
4: Concatenate pseudo-random sequence.
$S_R = concatenate(S_1, S_3)$, $S_G = concatenate(S_2, S_4)$, $S_B = concatenate(S_5, S_6)$
5: Reshape the sequence S_R, S_G, S_B of size [1×MN] into [M, N]
$S_R = reshape(S_R, [M, N])$, $S_G = reshape(S_G, [M, N])$, $S_B = reshape(S_B, [M, N])$
6: The matrix S_R, S_G, S_G perform quantization operation as shown in Eq. (7) and Eq. (8) to obtain matrix which is used for permutation and diffusion.

$$\begin{cases} K_{diff}(:,:,1) = mod(floor(mod(S_R, 10^{-3}) \times 10^{12}), 256) \\ K_{diff}(:,:,2) = mod(floor(mod(S_G, 10^{-3}) \times 10^{12}), 256) \\ K_{diff}(:,:,3) = mod(floor(mod(S_B, 10^{-3}) \times 10^{12}), 256), \end{cases} \tag{7}$$

$$\begin{cases} K_{comp}(:,:,1) = mod(floor(S_R \times 10^3), 3) \\ K_{comp}(:,:,2) = mod(floor(S_G \times 10^3), 3) \\ K_{comp}(:,:,3) = mod(floor(S_B \times 10^3), 3) \end{cases} \tag{8}$$

$$\begin{cases} C^G_{i,j} = P^G_{LS(i,j)+1,LS(i,j+1)+1} \times C^G_{i-1,N} + K_{diff}(i, j, 2) \\ j = 1 \\ C^G_{i,j} = P^G_{LS(i,j)+1,LS(i,j+1)+1} \times C^G_{i,j-1} + K_{diff}(i, j, 2) \\ j \neq 1 \\ C^G_{i,j} = P^G_{LS(i,j)+1,LS(i,j)+1} \times C^G_{i-1,j} + K_{diff}(i, j, 2) \\ j = N, \end{cases} \tag{16}$$

$$\begin{cases} C^B_{i,j} = P^B_{LS(i,j)+1,LS(i,j+1)+1} \times C^B_{i-1,N} + K_{diff}(i, j, 3) \\ j = 1 \\ C^B_{i,j} = P^B_{LS(i,j)+1,LS(i,j+1)+1} \times C^B_{i,j-1} + K_{diff}(i, j, 3) \\ j \neq 1 \\ C^B_{i,j} = P^B_{LS(i,j)+1,LS(i,j)+1} \times C^B_{i-1,j} + K_{diff}(i, j, 3) \\ j = N. \end{cases} \tag{17}$$

3) Repeat (3) of step 2 to permute among channels R, G and B.
4) Repeat (4) of step 2 to replace the pixels of channels R, G and B with boxes *Sbox_r*, *Sbox_g* and *Sbox_b* respectively.

Algorithm 3 Permutation

Input: image $C(i, j, 1)$, $C(i, j, 2)$, $C(i, j, 3)$, K_{comp}, S_R, S_G, S_B.
Output: permutated image $C(i, j, 1)$, $C(i, j, 2)$, $C(i, j, 3)$.

1: Initialize KT (:,:,:1)= S_R; KT (:,:,:2)= S_G;
KT (:,:,:3)= S_B
2: **for** k←1 to 3 **do**
3: **if** (k+K_{comp}(i, j, k))≥ 4 **then**
4:

$$t_i \leftarrow mod(floor\left(KT(i, j, k)\times 10^6\right), i) + 1; \tag{18}$$

5:

$$t_j \leftarrow mod(floor\left(KT(i, j, k)\times 10^6\right), j) + 1; \tag{19}$$

5: exchange (C (i, j, k), C (t_i, t_j,mod(k+K_{comp} (i, j, k), 3) +1));
6: **else**
7: exchange (C (i, j, k), C (t_i, t_j, k+K_{comp} (i, j, k)));
8: **end if**
9: **end for**

Step 4: Choose the next pixel value from left to right and from top to bottom, if $i \leq M, j \leq N$, perform Step 3, else perform step 5.

Step 5: Cipher image C is obtained. Where $C(i, j, 1) = C^R$, $C(i, j, 2) = C^G$, $C(i, j, 3) = C^B$.

IV. SIMULATION RESULTS

Our algorithm is implemented on MATLAB R2014a platform on a PC with an Intel (R) Core (TM) i5-6500 CPU 3.2GHz, 8GB RAM, 1TB hard disk and Windows 7 64-bit operating system. The color plain-images "Lena", "Peppers", "Baboon", "Boats" with size 512 × 512×3 are used for encryption. In our encryption algorithm, the keys used include: initial values $x_1(1), x_2(1), x_3(1), x_4(1), x_5(1), x_6(1)$ and parameters ε_0, μ_0 of chaotic system. The 256-bit hash value K.

A. KEY SPACE ANALYSIS

In order to ensure that the cryptosystem can withstand brute force attack, the key space of the cryptosystem should not be less than $2^{100}(\approx 10^{30})$. In our image encryption algorithm, the key is a 256-bit hash value K, and $x_1(1), x_2(1), x_3(1), x_4(1), x_5(1), x_6(1), \varepsilon_0, \mu_0$. If the accuracy of the computer is 10^{-14}. The key space will be $S_{key} = 10^{14\times 8} \times 2^{256} \approx 2^{656}$.So the key space size is greater than 2^{100}. Therefore, the key space is large enough to resist brute force attacks.

B. HISTOGRAM ANALYSIS

Histogram describes the distribution of image pixel values. If the distribution is not uniform, the attacker can be prevented from obtaining the statistical information of cipher image through statistical analysis, so histogram is often used to evaluate the performance of cryptosystem. Fig. 4 is the

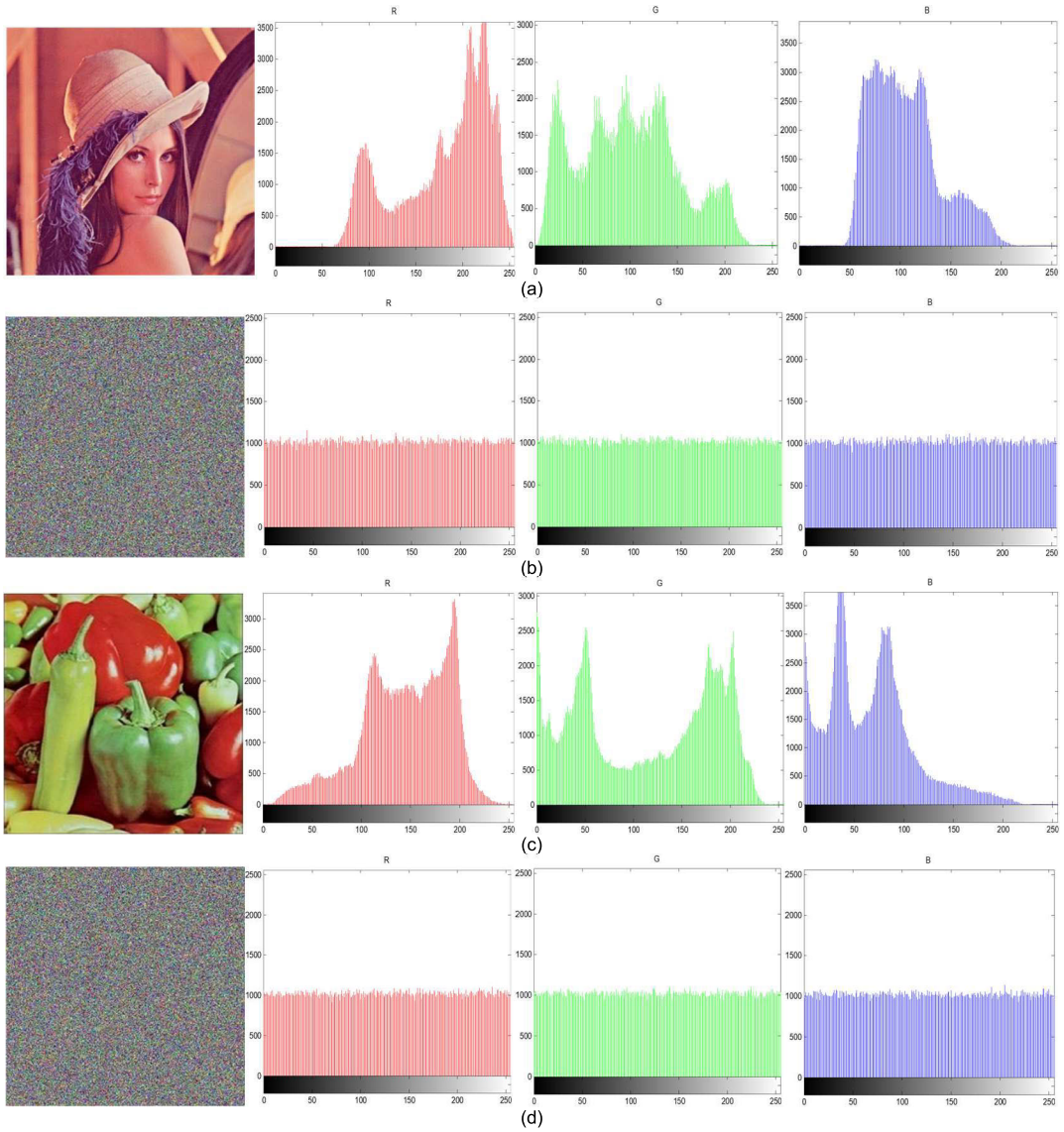

FIGURE 4. Histograms of plain image and cipher image: (a) Histogram of plain image "Lena", (b) Histogram of cipher image "Lena", (c) Histogram of plain image "Peppers", (d) Histogram of cipher image "Peppers".

results of plain image and cipher image. It can be seen that compared with plain image, histogram of cipher image is evenly distributed and can resist statistical attack.

In order to further evaluate the uniformity of ciphertext histogram, the variance defined in Eq. (20) is used to evaluate the uniformity of histogram. The smaller the variance, the higher the uniformity.

$$Var = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\left(x_i - x_j\right)^2}{2}. \tag{20}$$

where n is the number of gray levels, x_i and x_j is the number of pixels with corresponding gray values of i and j, respectively. For Lena ($512 \times 512 \times 3$), the variance of histogram of plain image and cipher image is shown in Table 3. It can be seen from the calculation results that compared with the corresponding values of plain images, the variance of the algorithm in this paper is greatly reduced and smaller than that of the existing algorithm, which is effective.

C. INFORMATION ENTROPY ANALYSIS

Information entropy is a measure of information randomness, which can be calculated by follows:

$$H(m) = -\sum_{i=0}^{255} P(m_i)\, log_2 P(m_i), \tag{21}$$

For a source image with 256 gray levels, each gray level has 8 bits. When the probability of each gray level is equal, the encrypted image can reach the ideal entropy of 8, namely, each gray level of the encrypted image is evenly distributed. The larger the entropy, and the less information it contains, the more chaotic the image.

TABLE 3. The variance analysis of R, G, B components and the variance comparison results of "Lena" images.

Algo	Image	Variances						
		Plain image			Cipher image			
		R	G	B	R	G	B	avg
Ours	Lena	1,017,335	4,557,188	1,377,356	948.4141	966.5391	978.7656	964.5729
	Peppers	8,191,10	4,366,90	1,196,978	968.0625	942.8828	954.7266	955.224
	Baboon	3,083,46	5,586,38	3,173,64	976.4609	874.9453	1006.8	952.7354
	Boats	9,717,57	1,616,274	2,193,472	1206.02	842.4922	1102.1	1050.204
	Fruits	5,777,41	8,426,23	1,880,277	1005.3	886.1719	1013.84	968.4896
	Barbara	3,165,978	5,531,75	4,881,55	1085.8	1000.9	1023.1	1036.6
Ref. [31]	Lena	1,017,30	455,720	1,377,40	940.4063	972.8125	995.5	969.5729
Ref. [32]	Lena	1,017,30	455,720	1,377,40	895.14	1112	1050.4	1019.18
Ref. [33]	Lena		-		1070	995.320	995.828	1020.383
Ref. [34]	Lena			-				977.02
Ref. [35]	Lena			-				974.8

TABLE 4. Entropies of plain images and cipher images.

Image	Entropy					
	Plain image			Cipher image		
	R	G	B	R	G	B
Lena	7.2531	7.5940	6.9684	7.9994	7.9993	7.9993
Peppers	7.3827	7.6631	7.1987	7.9993	7.9992	7.9992
Baboon	7.7324	7.4826	7.7570	7.9994	7.9993	7.9992
Boats	7.4208	7.0850	7.0315	7.9993	7.9994	7.9993
Fruits	7.5344	7.3320	6.8292	7.9993	7.9994	7.9993
Barbara	7.7023	7.5190	7.5850	7.9993	7.9993	7.9993

TABLE 5. The comparison of correlation coefficients of plain image and cipher image.

Algorithm	Image		Plain image			Cipher image		
			R	G	B	R	G	B
Ours	Lena	Horizontal	0.9898	0.9829	0.9580	-0.0055	0.0185	0.0018
		Vertical	0.9799	0.9696	0.9345	-0.0036	-0.00008	-0.0008
		Diagonal	0.9682	0.9541	0.9151	-0.0069	-0.0026	-0.0044
	Peppers	Horizontal	0.9879	0.9946	0.9883	0.0081	-0.0118	0.0103
		Vertical	0.9888	0.9942	0.9879	0.0173	-0.0159	-0.0260
		Diagonal	0.9774	0.9893	0.9776	-0.0130	0.0103	0.0088
	Baboon	Horizontal	0.8688	0.7838	0.8836	-0.0024	-0.0135	0.0052
		Vertical	0.9245	0.8776	0.9247	0.0013	-0.0049	0.0245
		Diagonal	0.8505	0.7386	0.8513	0.0049	-0.0010	0.0064
	Boats	Horizontal	0.9739	0.9785	0.9810	0.0043	-0.0061	0.0019
		Vertical	0.9639	0.9669	0.9718	0.0109	0.0154	0.0095
		Diagonal	0.9389	0.9470	0.9545	-0.0121	0.0068	0.0008
	Fruits	Horizontal	0.9923	0.9829	0.9580	0.0045	0.0077	0.0087
		Vertical	0.9929	0.9696	0.9345	-0.0134	0.0035	-0.0010
		Diagonal	0.9856	0.9541	0.9151	0.0015	0.0038	0.0044
	Barbara	Horizontal	0.9617	0.9592	0.9640	0.0045	0.0077	0.0087
		Vertical	0.8883	0.8704	0.8914	-0.0098	-0.0128	-0.0141
		Diagonal	0.8691	0.8523	0.8747	-0.0092	0.0079	0.0051
Ref. [36]	Lena	Horizontal	0.9813	0.9691	0.9455	0.0034	-0.0014	-0.0173
		Vertical	0.9803	0.9594	0.9294	0.0203	-0.0025	0.0006
		Diagonal	0.9668	0.9433	0.9099	-0.0073	-0.0131	0.0111
Ref. [37]	Lena	Horizontal	0.9757	0.9671	0.9472	-0.0127	-0.0075	-0.0007
		Vertical	0.9521	0.9325	0.9194	0.0067	-0.0068	0.0042
		Diagonal	0.9334	0.9128	0.8842	0.0060	-0.0078	0.0026
Ref. [38]	Lena	Horizontal	0.9244	0.9336	0.8545	0.0137	-0.0246	-0.0137
		Vertical	0.9765	0.9794	0.9498	-0.0237	-0.0170	0.0023
		Diagonal	0.9366	0.9368	0.9046	0.0109	-0.0133	-0.0013
Ref. [39]	Lena	Horizontal	0.9788	0.9686	0.9289	-0.0032	-0.0076	0.0240
		Vertical	0.9897	0.9820	0.9580	0.0141	-0.0060	-0.0084
		Diagonal	0.9725	0.9688	0.9205	0.0058	-0.0019	0.0065
Ref. [40]	Lena	Horizontal	-	-	-	0.0035	-0.0097	0.0186
		Vertical	-	-	-	-0.0041	0.0053	0.0101
		Diagonal	-	-	-	0.0410	-0.0085	-0.0175

Table 4 shows the information entropy of plain images and cipher images. It can be seen from the table that the information entropy of cipher images is close to 8. Therefore, cipher images have a better random distribution and the possibility of information leakage is 0.

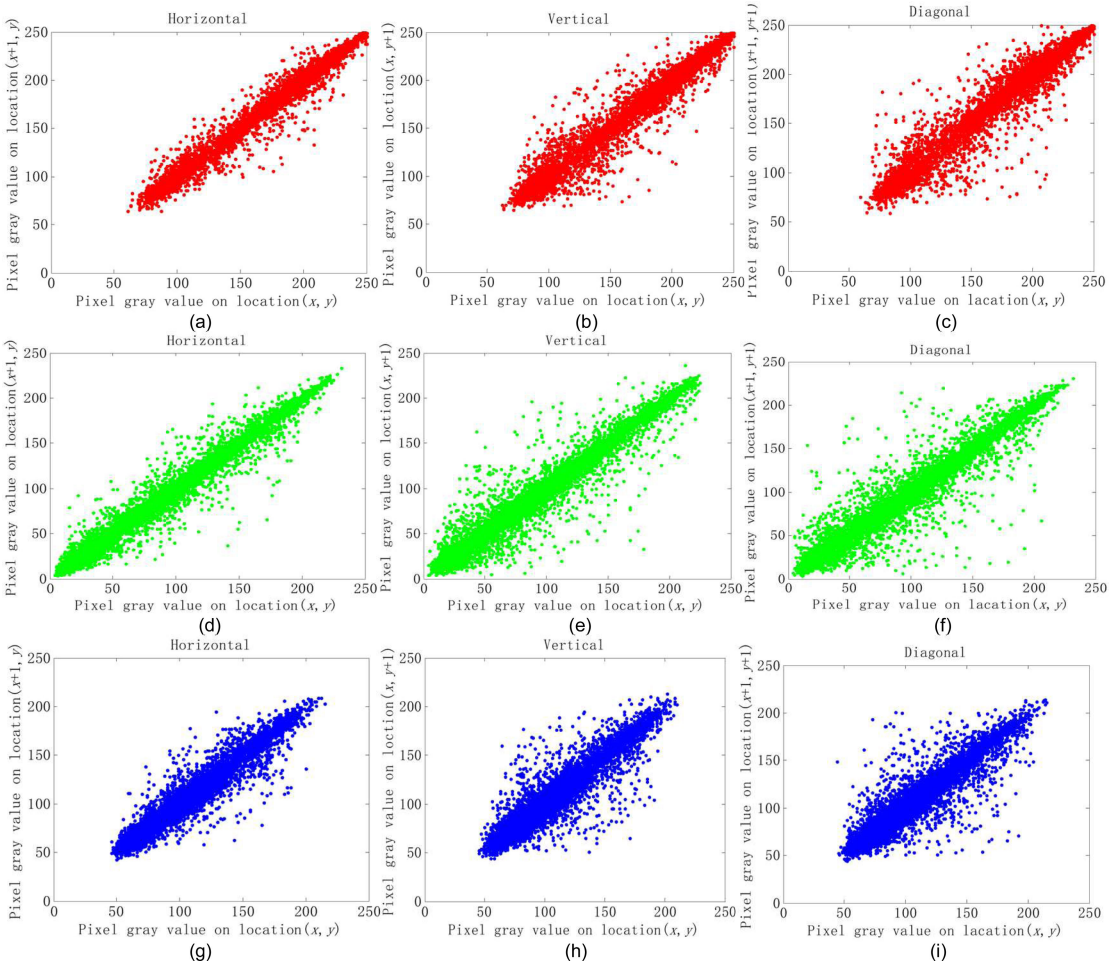

FIGURE 5. Correlation of adjacent pixels of R, G, B channels of plain image Lena: (a)-(c) Correlation in horizontal, vertical and diagonal directions of R channel, (d)-(f) Correlation in horizontal, vertical and diagonal directions of G channel, (g)-(i) Correlation in horizontal, vertical and diagonal directions of B channel.

D. CORRELATION ANALYSIS

The adjacent pixels of plain image have a strong correlation, and a good encryption algorithm should be able to greatly reduce this correlation. The correlation of adjacent pixels is calculated by Eq. (22)-(25)

$$r_{xy} = \frac{|cov(x, y)|}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{22}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2, \tag{23}$$

$$cov(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)), \tag{24}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i. \tag{25}$$

where x and y are the pixel values of two neighboring pixels in the image, r_{xy} is the correlation coefficient of two variables. $r_{xy} \leq 1$ means that adjacent pixels have strong correlation. The smaller r_{xy} is, the lower the correlation coefficient of encrypted image is, so hackers can't get the useful encrypted information. In order to analyze the correlation between adjacent pixels of plaintext and ciphertext, 8000 pairs of adjacent pixels are selected from horizontal, vertical and diagonal directions to draw the distribution diagram. It can be seen from Fig. 5 that in all directions, the correlation coefficient of plain image is higher, which is close to 1, whereas that of cipher image is lower as shown in Fig. 6.

Table 5 lists the quantitative analysis of the correlation of different channels. It can be seen that the correlation coefficients of different channels in different directions are close to 0. Moreover, compared with some existing algorithms, the proposed algorithm has better correlation. Therefore, the proposed algorithm can better resist statistical attacks.

E. DIFFERENTIAL ATTACK ANALYSIS

Attackers usually make small changes to the plain image, then use the same encryption algorithm to encrypt the plain image and the modified plain image, and compare the two cipher images to further find out the relationship between the plain image and the cipher image. In order to test the effect of slightly change of plain image on the corresponding cipher image, number of pixels change rate (NPCR) and unified average change intensity (UACI) are proposed to measure the

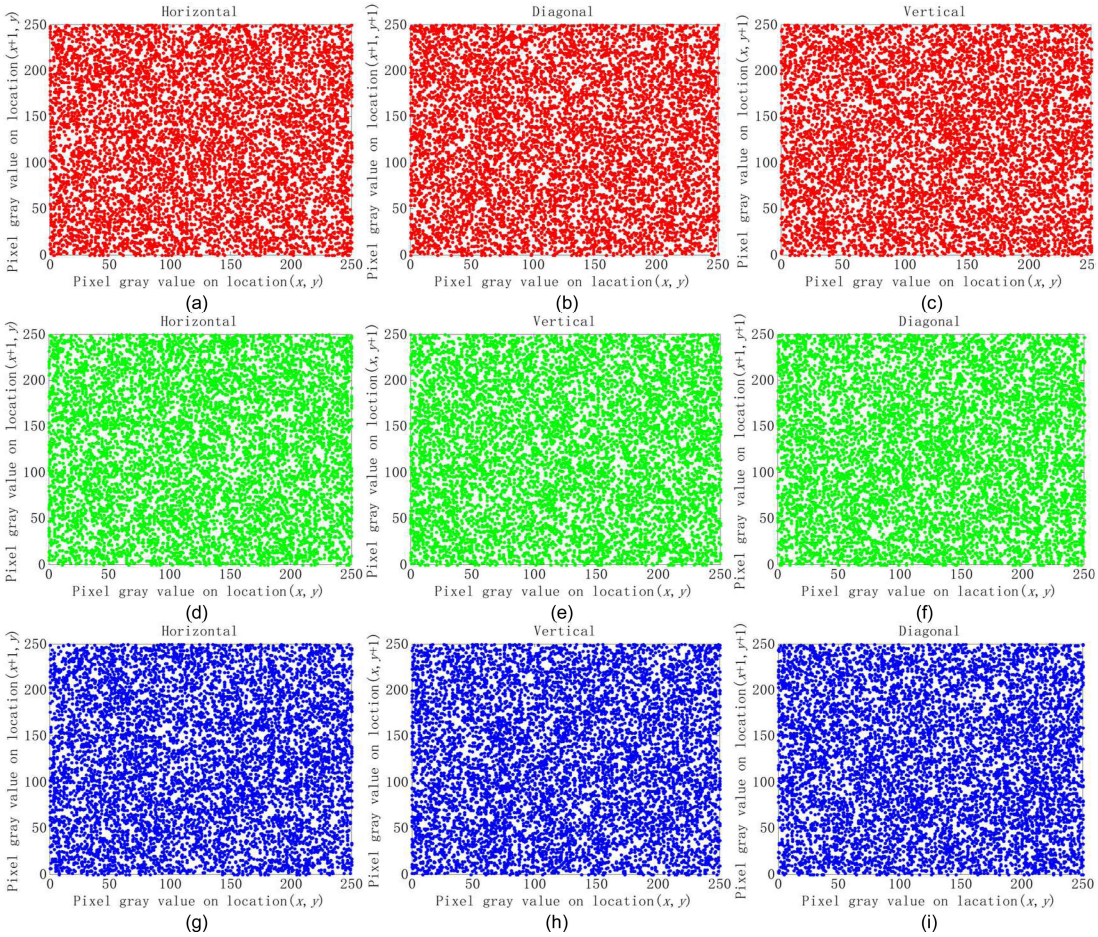

FIGURE 6. Correlation of adjacent pixels of R, G, B channels of cipher image Lena: (a)-(c) Correlation in horizontal, vertical and diagonal directions of R channel, (d)-(f) Correlation in horizontal, vertical and diagonal directions of G channel, (g)-(i) Correlation in horizontal, vertical and diagonal directions of B channel.

TABLE 6. The results of NPCRs and UACIs of various image in the proposed scheme.

Image	NPCR				UACI			
	R	G	B	Average	R	G	B	Average
Lena	99.5979	99.6258	99.6101	99.6113	33.5131	33.4958	33.4672	33.4920
Peppers	99.6132	99.6334	99.6281	99.6249	33.4435	33.5009	33.3976	33.4477
Baboon	99.5995	99.6262	99.6304	99.6187	33.5411	33.4003	33.4232	33.4549
Boats	99.6212	99.6178	99.6059	99.6150	33.4625	33.4603	33.4463	33.4564
Fruits	99.6052	99.6143	99.6113	99.6112	33.4292	33.4437	33.4915	33.4359
Barbara	99.5945	99.6262	99.61128	99.6106	33.4322	33.4558	33.4915	33.4599

anti-differential attack performance of the encryption algorithm, the calculation for NPCR and UACI are as follows:

$$\begin{cases} NPCR = \dfrac{\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j)}{M \times N} \times 100\% \\ D(i,j) = \begin{cases} 1, C_1(i,j) \neq C_2(i,j) \\ 0, C_1(i,j) = C_2(i,j), \end{cases} \end{cases} \tag{26}$$

$$UACI = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} |C_1(i,j) - C_2(i,j)|}{M \times N} \times 100\%. \tag{27}$$

where M and N are the height and width of the cipher image, C_1, C_2 are two cipher images with one-bit difference on responding plain images. As stated in Ref. [41], the expected values of *NPCR* and *UACI* for a true color image is $NPCR_R = NPCR_G = NPCR_B = 99.6094\%$ and $UACI_R = UACI_G = UACI_B = 33.4635\%$.

In the simulation, we randomly choose a pixel to modify its value, then encrypt the image before and after the change to get two cipher images, and calculate the NPCR and UACI of R, G and B channel by using Eq. (25), Eq. (26) and calculate their respective mean values. The results of NPCRs and UACIs are shown in Table 6. From Table 6, it can be concluded that the average $NPCR_{R,G,B}$ of the encrypted image is more than 99.6094% and average $UACI_{R,G,B}$ is more

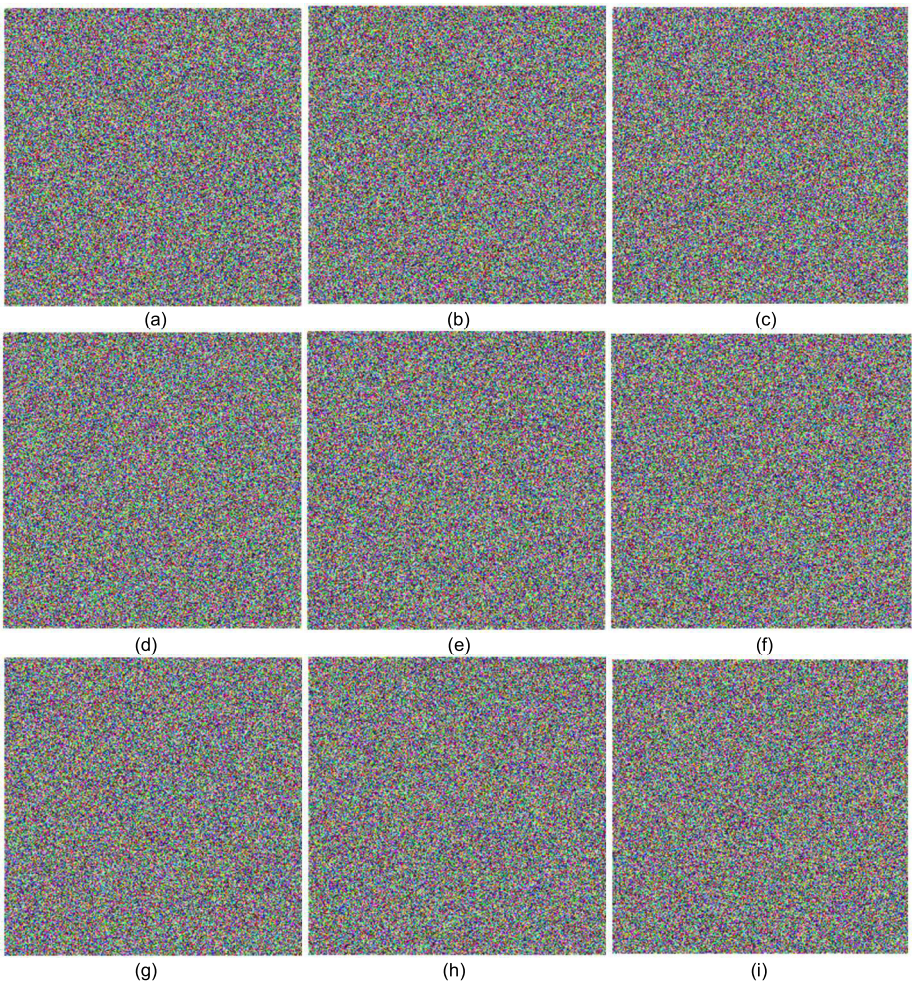

FIGURE 7. Decrypted image with wrong key: (a) Decrypted image with $\varepsilon_0 + 10^{14}$, (b) Decrypted image with $\mu_0 + 10^{14}$, (c) Decrypted image with x_1 (1), (d) Decrypted image with x_2 (1), (e) Decrypted image with x_3 (1), (f) Decrypted image with x_4 (1), (g) Decrypted image with x_5 (1), (h) Decrypted image with x_6 (1).

TABLE 7. The comparison of NPCR and UACI values of Lena.

Item	Ours	Ref. [32]	Ref. [33]	Ref. [42]	Ref. [43]	Ref. [44]
Average $NPCR_{R,G,B}$	99.61	99.61	99.60	99.61	99.61	99.62
Average $UACI_{R,G,B}$	33.49	33.46	33.32	33.45	33.46	33.48

than 33.4635%, which is very close to the expected value. Therefore, the encryption algorithm is plaintext sensitive and can resist differential attacks. Table 7 shows the comparison between the proposed encryption algorithm and some existing algorithms, which shows that the proposed algorithm is more effective.

F. KEY SENSITIVITY ANALYSIS

1) KEY SENSITIVITY ANALYSIS OF ENCRYPTION PROCESS

In order to test the key sensitivity of the encryption process, the Lena image with the size of $256 \times 256 \times 3$ was chosen as the test image. Change a key in the process of encryption and remain other key unchanged. For sake of testing the sensitivity of the hash value K, K is modified to K', K and K' are as follows:

$$K = \begin{bmatrix} \mathbf{73\ 30\ 39\ 73\ 13\ 69\ 253\ 241\ 149\ 7\ 135\ 241\ 109\ 0146\ 193} \\ 53\ 75\ 210\ 176\ 162\ 199\ 80\ 19\ 7\ 0\ 93\ 77\ 11\ 188\ 64\ 126 \end{bmatrix}$$

$$\downarrow$$

$$K' = \begin{bmatrix} \mathbf{74\ 30\ 39\ 73\ 13\ 69\ 253\ 241\ 149\ 7\ 135\ 241\ 109\ 0146\ 193} \\ 53\ 75\ 210\ 176\ 162\ 199\ 80\ 19\ 7\ 0\ 93\ 77\ 11\ 188\ 64\ 126 \end{bmatrix}$$

Table 8 is the result of quantitative analysis of key sensitivity. It can be seen from Table 8 that when the key changes slightly, the encrypted image is completely different. The difference

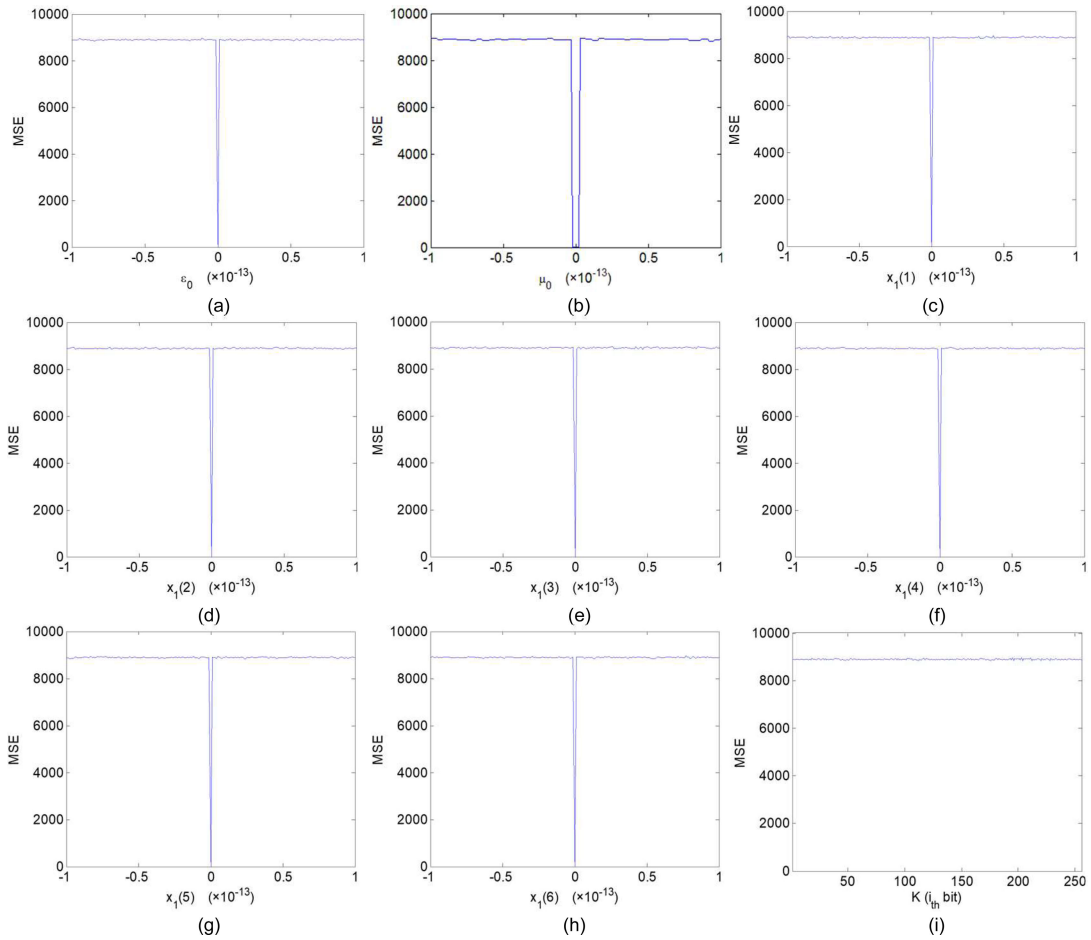

FIGURE 8. MSE curves: (a) ε_0, (b) μ_0, (c) x_1 (1), (d) x_2 (1), (e) x_3 (1), (f) x_4 (1), (g) x_5 (1), (h) x_6 (1).

TABLE 8. Analysis results of key sensitivity in the encryption process of Lena.

	R		G		B	
EK	NPCR	UACI	NPCR	UACI	NPCR	UACI
$\varepsilon + 10^{-14}$	99.6094	33.4421	99.6155	33.4846	99.5483	33.3945
$\mu + 10^{-14}$	99.6124	33.4654	99.5834	33.5187	99.5895	33.5156
$x_1(1) + 10^{-14}$	99.6353	33.5135	99.5941	33.5079	99.5926	33.6430
$x_2(1) + 10^{-14}$	99.5682	33.4134	99.6216	33.4059	99.6368	33.4925
$x_3(1) + 10^{-14}$	99.5651	33.3493	99.5926	33.5647	99.6078	33.4453
$x_4(1) + 10^{-14}$	99.6384	33.5688	99.6216	33.3875	99.5667	33.4018
$x_5(1) + 10^{-14}$	99.5911	33.5772	99.5758	33.5758	99.6109	33.4897
$x_6(1) + 10^{-14}$	99.6185	33.3863	99.5956	33.4540	99.6155	33.3097
K_1	99.6155	33.4365	99.6094	33.5904	99.5667	33.5271

rate between the encrypted image with the correct key and the encrypted image with the wrong key is greater than 99.60%, which reflects the high sensitivity of the proposed algorithm in the encryption process.

Fig. 7 is the decryption results decrypted with the wrong key. It can be seen from Fig. 7 that when the key is slightly changed, the decrypted image is like noise, and no useful information can be obtained. Fig. 8(a) - (c) shows an MSE curve of plain image and decrypted image with an incorrect key. Fig. 8 (a) abscissa indicates that the key ε_0 changes 10^{-13}, and the other keys remain unchanged. Fig. 8 (b) abscissa indicates that the key μ_0 changes 10^{-13}, and the other keys remain unchanged. Fig. 8 (c)-(h) abscissa indicates that the key x_1 (1), x_1 (2), x_1 (3), x_1 (4), x_1 (5), x_1 (6) changes 10^{-13}, and the other keys remain unchanged. Fig. 8 (d) is the MSE curve obtained when the ith bit of key K is reversed, the other keys remain unchanged. It is obvious that the MSE value is very large when one key for the decryption process has a tiny deviation and the others remain unchanged.

TABLE 9. The contrast analysis of binary images and corresponding cipher images.

	Image	Lena	Peppers	Baboon	Boats	Average for each component
Plain	R	0.1767	0.1039	0.5127	0.2752	0.2671
	G	0.2554	0.1183	0.5704	0.2544	0.2996
	B	0.2404	0.1002	0.5666	0.2400	0.2868
	Average for all images				0.2845	
Cipher	R	8.5892	8.5930	8.5895	8.6236	8.5988
	G	8.5985	8.6108	8.5493	8.5770	8.5839
	B	8.5671	8.5835	8.5715	8.5735	8.5739
	Average for all images				8.5855	

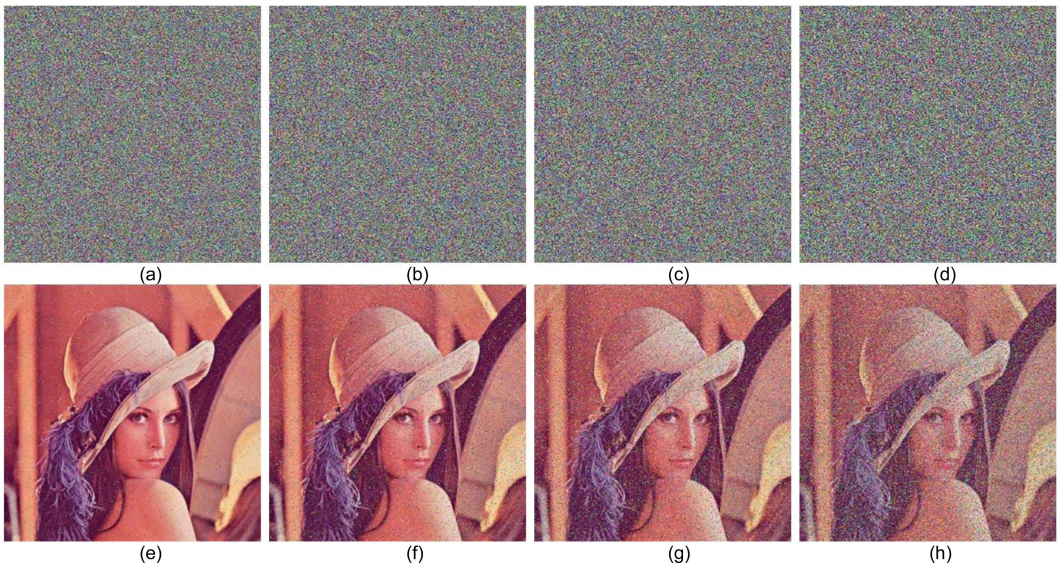

FIGURE 9. Cipher and decryption of SPN noise with different densities: (a) Encrypted image with noise density 0.01, (b) Encrypted image with noise density 0.05, (c) Encrypted image with noise density 0.1, (d) Encrypted image with noise density 0.2, (e) decrypted image from (a), (f) decrypted image from (b), (g) decrypted image from (c), (h) decrypted image from (d).

TABLE 10. Comparision of time complxity of different algorithms.

Algorithms	Time complexity
The proposed algorithm	$\boldsymbol{\Theta(6 \times M \times N)}$
Ref. [46]	$\boldsymbol{\Theta(6 \times M \times N)}$
Ref. [47]	$\boldsymbol{\Theta(168 \times M \times N)}$
Ref. [48]	$\boldsymbol{\Theta(8 \times M \times N)}$
Ref. [49]	$\boldsymbol{\Theta(9 \times M \times N)}$
Ref. [50]	$\boldsymbol{\Theta(69 \times M \times N)}$

G. ROBUSTNESS ANALYSIS

Image will inevitably be perturbated by various kinds of noise during the network transmission, such as salt and pepper noise, Gaussian noise and so on. Therefore, a good encryption algorithm should have good robustness to a certain extent.

1) NOISE ATTACK

In this simulation, we add salt and pepper noise (SPN) and Gaussian noise (GN) of different density to the Lena cipher image with the size of $512 \times 512 \times 3$. Fig. 9 (a)-(d) are the cipher images with the SPN noise density of 0.01, 0.05, 0.1 and 0.2, respectively. Fig. 9 (e)-(h) are corresponding decrypted images. Fig. 10(a)-(d) are cipher images with GN noise density of 10^{-6}, 2×10^{-6}, 3×10^{-6} and 5×10^{-6}, respectively, and Fig. 10(e)-(h) are corresponding decrypted image. It can be seen from the image that with the increase of noise density, more noise points will be found in the decrypted image. When the SPN density reaches 0.2 and the GN reaches 5×10^{-6}, the decrypted image can still be distinguishable, which shows that the proposed algorithm can resist the noise attack.

2) CROPPING ATTACK

When digital images are transmitted on the Internet, some information will be lost due to network congestion or malicious damage of attackers. Therefore, it is necessary for the encryption scheme to resist the cropping attack. In this simulation, as shown in Fig. 11 (a)-(d), we occlude 1 / 4, 16 / 1 and 1 / 2 of Lena encrypted image with the size of $512 \times 512 \times 3$,

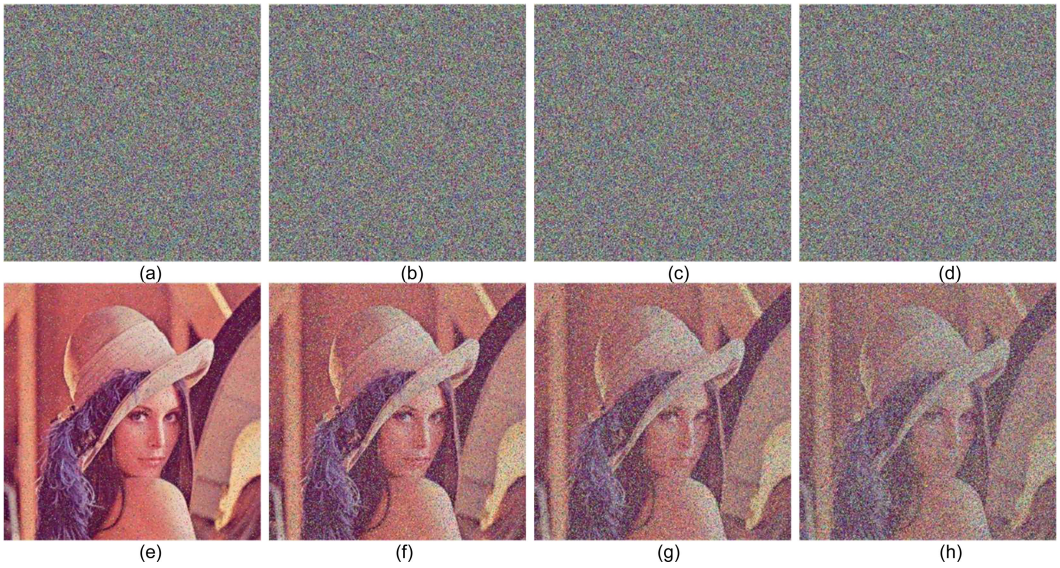

FIGURE 10. Cipher and decryption of GN noise with different densities: (a) Encrypted image with noise density 10^{-6}, (b) Encrypted image with noise density 2×10^{-6}, (c) Encrypted image with noise density 3×10^{-6}, (d) Encrypted image with noise density 5×10^{-6}, (e) decrypted image from (a), decrypted image from (b), (f) decrypted image from (c), (g) decrypted image from (c), (h) decrypted image from (d).

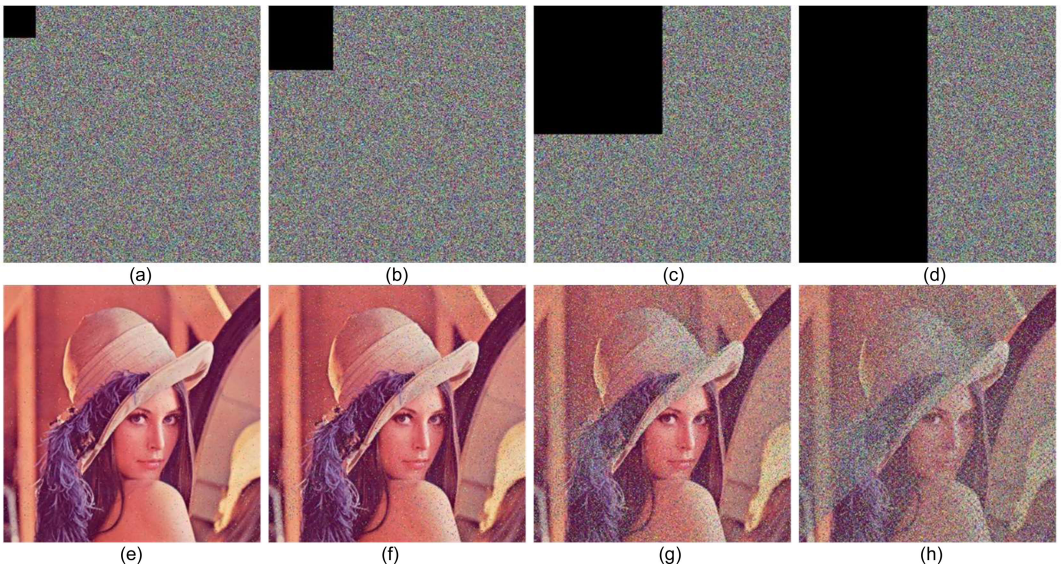

FIGURE 11. Cipher and decryption of cropping attack with percentages data loss: (a) Encrypted image with 1/64 data loss, (b) Encrypted image with noise density 1/16 data loss, (c) Encrypted image with noise density 1/4 data loss, (d) Encrypted image with noise density 1/2 data loss, (e) decrypted image from (a), (f) decrypted image from (b), (g) decrypted image from (c), (h) decrypted image from (d).

respectively. It can be seen from the Fig. 11(e)-(h) that with the increase of occlusion size, the quality of image restoration is decreasing, but the main information of the image can still be distinguished. Therefore, the proposed encryption scheme can resist cropping attack.

H. CONTRAST ANALYSIS

The intensity difference between pixels and their neighboring pixels in the whole image can be calculated by contrast analysis. High contrast indicates that the texture is not homogenous and the encryption effect is better. [23], [45]. The contrast is defined by Eq. (28)

$$C = \sum_{i,j=1}^{N} |\text{i-j}|^2 \, p\,(i,j), \tag{28}$$

where $p\,(i,j)$ is the number of gray-level co-occurrence matrices (GLCM), N is the number of rows and columns. The contrast analysis results of cipher images are shown in Table 9, which indicated that the proposed scheme has good contrast levels.

I. TIME COMPLEXITY ANALYSIS

In order to analyze the time cost of proposed encryption algorithm, we analyze the time complexity of the algorithm. In this paper, the time-consuming of the algorithm includes the generation of the key stream and the process of alternative encryption structure between permutation, diffusion and substitution. In the encryption process, NACML generates a chaotic sequence with a length of $\mathbf{3} \times \mathbf{M} \times \mathbf{N}$ and the time complexity is $\mathbf{\Theta}(\mathbf{3} \times \mathbf{M} \times \mathbf{N})$. In the encryption process, each pixel is encrypted only once. Namely, permutation, diffusion and substitution are carried out at the same time. So, the time complexity of the encryption process is $\mathbf{\Theta}(\mathbf{3} \times \mathbf{M} \times \mathbf{N})$. Therefore, the total time complexity is $\mathbf{\Theta}(\mathbf{6} \times \mathbf{M} \times \mathbf{N})$. It can be seen from Table 10 that the time complexity of the proposed algorithm is lower than that of some existing encryption algorithms. Therefore, it can be proved that our encryption algorithm is more effective.

V. CONCIUSION

In this paper, an alternative encryption structure between permutation, diffusion and substitution is proposed, which only encrypt every pixel once. Compared with the traditional encryption structure, which is implemented separately by permutation and diffusion, the proposed mechanism avoids the high algorithm complexity caused by multiple rounds of encryption, reduces the complexity and improves the security performance. Moreover, the combination of 3D Arnold transform and CML enlarges the key space, and NIST SP800-22 test shows that the generated random sequence has good randomness. Simulation results show that the algorithm can resist typical attacks and has good robustness and security. Simultaneously, the binary image can be encrypted only by making small changes to the algorithm. The modification is as follows: the original permutation between multiple channels is changed to that in a single channel. In the substitution process, only one S-box is constructed to substitute pixels, but due to the limited space, the experiment only encrypts and analyzes the encryption effect of the color image.

REFERENCES

[1] S. F. Raza and V. Satpute, "A novel bit permutation-based image encryption algorithm," *Nonlinear Dyn.*, vol. 95, no. 2, pp. 859–873, Jan. 2019.

[2] Y. Zhang, "The fast image encryption algorithm based on lifting scheme and chaos," *Inf. Sci.*, vol. 520, pp. 177–194, May 2020.

[3] S. Suri and R. Vijay, "A Pareto-optimal evolutionary approach of image encryption using coupled map lattice and DNA," *Neural Comput. Appl.*, Dec. 2019, doi: 10.1007/s00521-019-04668-x.

[4] A. Bisht, M. Dua, S. Dua, and P. Jaroli, "A color image encryption technique based on bit-level permutation and alternate logistic maps," *J. Intell. Syst.*, vol. 29, no. 1, pp. 1246–1260, Dec. 2019.

[5] H. Zhang, X.-Q. Wang, Y.-J. Sun, and X.-Y. Wang, "A novel method for lossless image compression and encryption based on LWT, SPIHT and cellular automata," *Signal Process., Image Commun.*, vol. 84, May 2020, Art. no. 115829.

[6] W. Yu, Y. Liu, L. Gong, M. Tian, and L. Tu, "Double-image encryption based on spatiotemporal chaos and DNA operations," *Multimedia Tools Appl.*, vol. 78, no. 14, pp. 20037–20064, Jul. 2019.

[7] Q. Xu, K. Sun, C. Cao, and C. Zhu, "A fast image encryption algorithm based on compressive sensing and hyperchaotic map," *Opt. Lasers Eng.*, vol. 121, pp. 203–214, Oct. 2019.

[8] F. Yang, J. Mou, K. Sun, Y. Cao, and J. Jin, "Color image compression-encryption algorithm based on fractional-order memristor chaotic circuit," *IEEE Access*, vol. 7, pp. 58751–58763, May 2019.

[9] M. Kaur, D. Singh, K. Sun, and U. Rawat, "Color image encryption using non-dominated sorting genetic algorithm with local chaotic search based 5D chaotic map," *Future Gener. Comput. Syst.*, vol. 107, pp. 333–350, Jun. 2020.

[10] Q. Xu, K. Sun, and C. Zhu, "A visually secure asymmetric image encryption scheme based on RSA algorithm and hyperchaotic map," *Phys. Scripta*, vol. 95, no. 3, Feb. 2020, Art. no. 035223.

[11] Y. Luo, R. Zhou, J. Liu, S. Qiu, and Y. Cao, "An efficient and self-adapting colour-image encryption algorithm based on chaos and interactions among multiple layers," *Multimedia Tools Appl.*, vol. 77, no. 20, pp. 26191–26217, Oct. 2018.

[12] D. D. Wheeler, "Problems with chaotic cryptosystems," *Cryptologia*, vol. 13, no. 3, pp. 243–250, Jul. 1989.

[13] X. Wang, Y. Wang, S. Wang, Y. Zhang, and X. Wu, "A novel pseudo-random coupled LP spatiotemporal chaos and its application in image encryption," *Chin. Phys. B*, vol. 27, no. 11, Nov. 2018, Art. no. 110502.

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

[15] Y.-J. Sun, H. Zhang, X.-Y. Wang, X.-Q. Wang, and P.-F. Yan, "2D non-adjacent coupled map lattice with q and its applications in image encryption," *Appl. Math. Comput.*, vol. 373, May 2020, Art. no. 125039.

[16] S. Guo, Y. Liu, L. Gong, W. Yu, and Y. Gong, "Bit-level image cryptosystem combining 2D hyper-chaos with a modified non-adjacent spatiotemporal chaos," *Multimedia Tools Appl.*, vol. 77, no. 16, pp. 21109–21130, Aug. 2018.

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

[18] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation Chaos*, vol. 08, no. 06, pp. 1259–1284, Jun. 1998.

[19] A. Arab, M. J. Rostami, and B. Ghavami, "An image encryption method based on chaos system and AES algorithm," *J. Supercomput.*, vol. 75, no. 10, pp. 6663–6682, Oct. 2019.

[20] R. I. Abdelfatah, "A new fast double-chaotic based Image encryption scheme," *Multimedia Tools Appl.*, vol. 79, nos. 1–2, pp. 1241–1259, Jan. 2020.

[21] S. El Assad and M. Farajallah, "A new chaos-based image encryption system," *Signal Process. Image Commun.*, vol. 41, pp. 147–157, Feb. 2016.

[22] C. Chen, K. Sun, and S. He, "An improved image encryption algorithm with finite computing precision," *Signal Process.*, vol. 168, Mar. 2020, Art. no. 107340.

[23] M. Dua, A. Wesanekar, V. Gupta, M. Bhola, and S. Dua, "Differential evolution optimization of intertwining logistic map-DNA based image encryption technique," *J. Ambient Intell. Humanized Comput.*, Nov. 2019, doi: 10.1007/s12652-019-01580-z.

[24] X. Wang, S. Wang, Y. Zhang, and C. Luo, "A one-time pad color image cryptosystem based on SHA-3 and multiple chaotic systems," *Opt. Lasers Eng.*, vol. 103, pp. 1–8, Apr. 2018.

[25] X. Kang, X. Luo, X. Zhang, and J. Jiang, "Homogenized Chebyshev-Arnold map and its application to color image encryption," *IEEE Access*, vol. 7, pp. 114459–114471, 2019.

[26] M. Zhang and X. Tong, "Joint image encryption and compression scheme based on IWT and SPIHT," *Opt. Lasers Eng.*, vol. 90, pp. 254–274, Mar. 2017.

[27] M. A. B. Farah, A. Farah, and T. Farah, "An image encryption scheme based on a new hybrid chaotic map and optimized substitution box," *Nonlinear Dyn.*, vol. 99, no. 4, pp. 3041–3064, Mar. 2020.

[28] E. Hasanzadeh and M. Yaghoobi, "A novel color image encryption algorithm based on substitution box and hyper-chaotic system with fractal keys," *Multimedia Tools Appl.*, vol. 79, nos. 11–12, pp. 7279–7297, Mar. 2020.

[29] U. Hayat and N. A. Azam, "A novel image encryption scheme based on an elliptic curve," *Signal Process.*, vol. 155, pp. 391–402, Feb. 2019.

[30] J. Zhou, J. Hu, and P. Chen, "Extended Euclid algorithm and its application in RSA," in *Proc. 2nd Int. Conf. Inf. Sci. Eng.*, Dec. 2010, pp. 2079–2081.

[31] K. A. Kumar Patro and B. Acharya, "An efficient colour image encryption scheme based on 1-D chaotic maps," *J. Inf. Secur. Appl.*, vol. 46, pp. 23–41, Jun. 2019.

[32] Z.-H. Gan, X.-L. Chai, D.-J. Han, and Y.-R. Chen, "A chaotic image encryption algorithm based on 3-D bit-plane permutation," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7111–7130, Nov. 2019.
[33] X.-L. Chai, Z.-H. Gan, Y. Lu, M.-H. Zhang, and Y.-R. Chen, "A novel color image encryption algorithm based on genetic recombination and the four-dimensional memristive hyperchaotic system," *Chin. Phys. B*, vol. 25, no. 10, Oct. 2016, Art. no. 100503.
[34] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using DNA sequence operations," *Opt. Lasers Eng.*, vol. 88, pp. 197–213, Jan. 2017.
[35] K. A. K. Patro, B. Acharya, and V. Nath, "Secure, lossless, and noise-resistive image encryption using chaos, hyper-chaos, and DNA sequence operation," *IETE Tech Rev.*, pp. 1–23, Apr. 2019, doi: 10.1080/02564602.2019.1595751.
[36] K. Xuejing and G. Zihui, "A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system," *Signal Process., Image Commun.*, vol. 80, Feb. 2020, Art. no. 115670.
[37] X. Wang and H.-L. Zhang, "A color image encryption with heterogeneous bit-permutation and correlated chaos," *Opt. Commun.*, vol. 342, pp. 51–60, May 2015.
[38] X. Wu, K. Wang, X. Wang, H. Kan, and J. Kurths, "Color image DNA encryption using NCA map-based CML and one-time keys," *Signal Process.*, vol. 148, pp. 272–287, Jul. 2018.
[39] Z. Xiong, Y. Wu, C. Ye, X. Zhang, and F. Xu, "Color image chaos encryption algorithm combining CRC and nine palace map," *Multimedia Tools Appl.*, vol. 78, no. 22, pp. 31035–31055, Nov. 2019.
[40] M. Kumar, A. Iqbal, and P. Kumar, "A new RGB image encryption algorithm based on DNA encoding and elliptic curve Diffie-Hellman cryptography," *Signal Process.*, vol. 125, pp. 187–202, Aug. 2016.
[41] X. Wu, D. Wang, J. Kurths, and H. Kan, "A novel lossless color image encryption scheme using 2D DWT and 6D hyperchaotic system," *Inf. Sci.*, vols. 349–350, pp. 137–153, Jul. 2016.
[42] M. L. Sahari and I. Boukemara, "A pseudo-random numbers generator based on a novel 3D chaotic map with an application to color image encryption," *Nonlinear Dyn.*, vol. 94, no. 1, pp. 723–744, Oct. 2018.
[43] M. Farajallah, S. El Assad, and O. Deforges, "Fast and secure chaos-based cryptosystem for images," *Int. J. Bifurcation Chaos*, vol. 26, no. 02, Feb. 2016, Art. no. 1650021.
[44] X. Ouyang, Y. Luo, J. Liu, L. Cao, and Y. Liu, "A color image encryption method based on memristive hyperchaotic system and DNA encryption," *Int. J. Mod. Phys. B*, vol. 34, no. 4, Feb. 2020, Art. no. 2050014.
[45] J. Ahmad and S. O. Hwang, "Chaos-based diffusion for highly autocorrelated data in encryption algorithms," *Nonlinear Dyn.*, vol. 82, no. 4, pp. 1839–1850, Dec. 2015.
[46] L. Huang, S. Wang, J. Xiang, and Y. Sun, "Chaotic color image encryption scheme using deoxyribonucleic acid (DNA) coding calculations and arithmetic over the Galois field," *Math. Problems Eng.*, vol. 2020, pp. 1–22, Mar. 2020.
[47] M. Samiullah, W. Aslam, H. Nazir, M. I. Lali, B. Shahzad, M. R. Mufti, and H. Afzal, "An image encryption scheme based on DNA computing and multiple chaotic systems," *IEEE Access*, vol. 8, pp. 25650–25663, 2020.
[48] C. Zhu, Z. Gan, Y. Lu, and X. Chai, "An image encryption algorithm based on 3-D DNA level permutation and substitution scheme," *Multimedia Tools Appl.*, vol. 79, nos. 11–12, pp. 7227–7258, Mar. 2020.
[49] G. Cheng, C. Wang, and H. Chen, "A novel color image encryption algorithm based on hyperchaotic system and permutation-diffusion architecture," *Int. J. Bifurcation Chaos*, vol. 29, no. 9, Aug. 2019, Art. no. 1950115.
[50] X. Wu, J. Kurths, and H. Kan, "A robust and lossless DNA encryption scheme for color images," *Multimedia Tools Appl.*, vol. 77, no. 10, pp. 12349–12376, May 2018.

HAO ZHANG (Member, IEEE) received the Ph.D. degree in computer software and theory from the Dalian University of Technology, China, in 2016. He is currently working with the College of Information and Computer, Taiyuan University of Technology, China. His research interests include systems biology, complex networks, and image processing.

XIAOQING WANG received the bachelor's degree in computer science and technology from Changzhi University, China, in 2017. She is currently pursuing the master's degree with the College of Information and Computer, Taiyuan University of Technology, China. Her research interests include image encryption and compression.

HONGWEI XIE received the Ph.D. degree in circuit and system from the Taiyuan University of Technology, China, in 2009. From 2009 to 2010, she was a Visiting Scholar with the Department of Computer Science, University of Auckland, New Zealand. From November 2016 to February 2017, she was a Professor for her research with the University of Ottawa, Canada. She is currently a Professor with the College of Software, Taiyuan University of Technology, China. She has published three books and more than 60 scientific articles in refereed journals and proceedings. Her research interests include artificial intelligence, image processing, and high performance computing.

CHUNPENG WANG (Member, IEEE) was born in Jinan, China, in 1989. He received the B.E. degree in computer science and technology from Shandong Jiaotong University, China, in 2010, the M.S. degree from the School of Computer and Information Technology, Liaoning Normal University, China, in 2013, and the Ph.D. degree from the School of Computer Science and Technology, Dalian University of Technology, China, in 2017. He is currently a Teacher with the School of Cyber Security, Qilu University of Technology (Shandong Academy of Sciences), China. His research interests include image processing and computer vision.

XINGYUAN WANG received the Ph.D. degree in computer software and theory from Northeast University, China, in 1999. From 1999 to 2001, he was a Postdoctoral Researcher with Northeast University. He is currently a Professor with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, China. He has published three books and more than 300 scientific articles in refereed journals and proceedings. His research interests include nonlinear dynamics and control, image processing, chaos cryptography, systems biology, and complex networks.