Signal Processing 171 (2020) 107484

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks

Minjun Zhou, Chunhua Wang*

College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China

ARTICLE INFO

Article history:
Received 8 November 2019
Revised 2 January 2020
Accepted 18 January 2020
Available online 20 January 2020

Keywords:
Image encryption
Conservative hyper-chaotic system
Closed-loop diffusion
Inter-block

ABSTRACT

Dissipative chaotic systems have been widely used in digital image encryption schemes in the past 20 years. However, compared to conservative chaotic systems, the dissipative chaotic systems have attractors thus attacker can reconstruct the chaotic systems by reconstructing the attractors. Therefore, the conservative chaotic systems are more suitable in chaos-based encryption system because they have no attractors thus can avoid the reconstructing attacks. Based on this, an image encryption scheme based on conservative hyper-chaotic system and closed-loop diffusion between blocks is proposed in this paper. On the one hand, the conservative hyperchaotic system has strong pseudo-randomness and anti-reconstruction attack property. On the other hand, different from the existing closed-loop diffusion schemes which change pixel values one by one until the last pixel value is changed, the proposed closed-loop diffusion method cannot only generate the first ciphertext and other ciphertext blocks but also update the first ciphertext block using the other generated ciphertext block information. In addition, the key streams are related to plaintext and ciphertext. Consequently, the key, plaintext and ciphertext form an organic whole to ensure the sensitivity of the encryption system. Moreover, simulation results and analysis show that the encryption scheme has strong security and excellent performance.

1. Introduction

With the development of modern communication technology and the popularity of the Internet, most media files such as text, images, and videos are exposed to a shared and accessible network space. Therefore, communication security in a digital world becomes a serious problem and gains more and more attention [1]. For instance, some images carrying bio-signals of fingerprints, irises, etc, can easily reveal important personal privacy and thus cause the leakage of personal privacy when they are transmitted on digital media without encryption. So the encryption of images is very necessary [2]. Thus, designing a secure digital image encryption method has also attracted the attention of many researchers. Different from one-dimensional text information, image data has its unique properties such as large amount of data, high redundancy, and strong correlation between adjacent pixels. These features make traditional encryption algorithms such as DES and AES for text encryption no longer suitable for image encryption [3,4].

Chaotic systems are widely used in image encryption schemes because of their unique properties such as pseudo-randomness, ergodicity, unpredictability, and sensitivity to initial values and parameters [5–10]. In 1998, Fridrich first proposed the basic framework of image encryption, which consisted of scrambling and diffusion [11]. In the scrambling stage, the position of the pixels in the image is scrambled, but the histogram of image doesn't be changed because the number of pixel values in image is invariable. While in the diffusion stage, the value of the pixels in the image can be changed, thus the histogram of the image can be changed. In recent decades, image encryption algorithms based on chaotic systems have appeared in large numbers [12–32]. Further, these encryption schemes can be divided into many categories such as schemes based on deoxyribonucleic acid (DNA) [12–16], block-based schemes [17–19], schemes based on complex chaotic systems [20–22], schemes based on special algorithms [23–27], schemes based on new scrambling diffusion structures [28–32] and so on.

Chaos occurs mainly in three categories of dynamic systems [33]: (a) conservative systems; (b) dissipative systems; and (c) quantum systems. Based on this, chaotic systems can be divided into dissipative chaotic system (DCS) and conservative chaotic system (CCS). CCS is a class of chaotic systems having the zero-sum of the Lyapunov exponents (LEs) [34] while DCS does not have this

* Corresponding author.
E-mail addresses: zmj0923@hnu.edu.cn (M. Zhou), wch1227164@hnu.edu.cn (C. Wang).

https://doi.org/10.1016/j.sigpro.2020.107484
0165-1684/© 2020 Elsevier B.V. All rights reserved.

property. DCSs are widely used in chaos-based image encryption algorithms, mainly because DCSs can realize hyperchaos easily and have strong pseudorandom behavior. These properties meet the requirements of big key space and high randomness for encryption system. For example, Zhang et al. [12] proposed a new image encryption scheme which was based on the spatiotemporal chaos of the Mixed Linear-Nonlinear Coupled Map Lattices (MLNCML) and the strategy of DNA computing. Wu et al. [15] described an image encryption algorithm which used two-dimensional Hénon-Sine map (2D-HSM) and DNA approach. Both Yin and Wang [23] and Li et al. [28] utilized hyperchaotic systems to generate the pseudo-random numbers which were used in the scrambling and diffusion stages. A dissipative chaotic system may have outstanding performance in pseudo-randomness, nevertheless it generates a strange attractor with fractional dimension, and its orbits approximately approach a manifold of fractional dimension having zero volume. Therefore, the risk that an attacker reconstructing the attractors and then cracking the encryption scheme is greatly increased, thus the image encryption schemes using the DCS are less secure. In addition, most of the space around the attractor is not reachable for orbit, hence the dissipative chaos are ergodicity-poor [33]. On the contrary, a conservative chaotic system does not produce any attractors, thus the attacker cannot decipher encryption algorithm by reconstructing the chaotic system used in the encryption algorithm, which can greatly improve the security and reliability of encryption scheme. Meanwhile, the dimension of a conservative chaotic system is an integer equaling to the system dimension, which brings about a richer ergodicity than the DCS [35]. However, up to now, the image encryption scheme based on conservative chaotic system has not been reported. Therefore, in this paper, we propose a new image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks. On the one hand, in contrast, the biggest Lyapunov exponent of the conservative hyper-chaotic system is larger than that of general conservative chaotic system, and the dynamic characteristics of conservative hyper-chaotic system are better, so the image encryption system using conservative hyperchaotic system is better in security and pseudo-randomness. On the other hand, compared with the open-loop encryption structure of the existing schemes [16–18], the closed-loop diffusion structure proposed in this paper greatly improves the reliability of encryption system.

In summary, an image encryption algorithm based on 5-dimensional conservative hyper-chaotic system and closed-loop diffusion between blocks is introduced. Firstly, the conservative hyper-chaotic system used in this paper has no attractor, and the biggest Lyapunov exponent is also very large. In consequence, it can resist the attacker from deciphering the encryption scheme by reconstructing the attractor. Additionally, the strong pseudo-randomness of the conservative hyper-chaotic system can further ensure the randomness and reliability of the encryption scheme. Secondly, the initial values of chaotic sequences used for scrambling and diffusion are generated in different ways. The initial values of chaotic sequences used for scrambling are constructed by using the SHA 256 hash of the plain image, while the initial values of chaotic sequences used for diffusion are quantified according to the size and the pixel values of plaintext. These ways make the key sequences for scrambling and diffusion related to plaintext, and they ensure that the encryption scheme can resist chosen-plaintext attacks. Finally, the block-based encryption scheme really realizes closed-loop diffusion. The generation of the current cipher block depends on the current plaintext block, a random cipher block, and the key block. The random cipher block locates before the current plaintext block, and the key block corresponds to the current cipher block. The generation of first cipher block depends on the plaintext block and the initial key block, but when the last ciphertext block is generated, the first cipher block is updated. The closed-loop diffusion structure makes the key block, the plaintext block and the cipher text block together form a closed-loop organic whole, so that the attacker cannot find the attack window and thus ensure the security of the encryption scheme.

The rest of the paper is organized as follows: in Section 2, we introduce the five-dimensional conservative hyper-chaotic system used in this work. Then the proposed encryption scheme and the decryption process are presented in Section 3. After that, the performance analysis of the encryption system is showed in Section 4, while Section 5 concludes the paper.

2. Preliminary works

2.1. 5D conservative hyper-chaotic system

Existing conservative chaotic systems can be divided into two categories. One is the Hamiltonian CCS (HCCS) [36–38], whose volume and total Hamiltonian are both conserved. The other one is non-Hamiltonian CCS (non-HCCS) [39–41], whose volume is conservative but total Hamiltonian is not conserved. Here we use a five-dimensional HCCS (5D-HCCS) in our image encryption scheme [42]. It can be described by the following Eq. (1).

$$\begin{cases} \dot{x} = ay + cv \\ \dot{y} = -ax + bxu \\ \dot{z} = du \\ \dot{u} = -bxy - dz \\ \dot{v} = -cx \end{cases} \tag{1}$$

where a, b, c and d are the parameters of the chaotic system. When we fix the parameters of the system (a, b, c, d) to (30, 30, 10, 30) and set its initial value formed by two negative numbers and three positive numbers, the system exhibits hyper-chaotic characteristics. For example, when we set the initial value of this system as (0.25, 7, −2, 2.5, −0.7), we can see from Fig. 1 that its LEs are (6.755464, 0.849765, 0.039421, −0.377339, −7.188469), which indicates that this system produces hyper-chaos and its pseudo-randomness is quite strong. As a result, this system is ideal as a random number generator for image encryption systems.

In order to compare the LEs of the five-dimensional conservative hyperchaotic system with the general conservative chaotic systems, we consider the following four-dimensional conservative chaotic system (4D-CCS) shown in Eq. (2) [35]. Meanwhile, Fig. 2 gives the distribution maps of the LEs of the 4D-CCS. It can be seen from Fig. 2 that when the initial values ($x0$, $y0$, $z0$, $w0$) of the 4D-CCS are set as (0.25, 7, −2, 2.5) and the parameters (a, b, c, d) of it are fixed as(30, 30, 10, 30), the LEs of it are (-0.000000, −0.000000, 0.391222, −0.391222), which shows that the 4D-CCS can only produce the chaotic phenomenon and its pseudo-randomness is not strong due to its biggest LE is close to 0. Therefore, the 4D-CCS is not suitable for generating a series of pseudo-random numbers required for image encryption systems.

$$\begin{cases} \dot{x} = (c-b) \times y \times z + (d-c) \times z \times w \\ \dot{y} = (a-c) \times x \times z \\ \dot{z} = (b-a) \times x \times y + (a-d) \times x \times w \\ \dot{w} = (c-a) \times x \times z \end{cases} \tag{2}$$

where a, b, c and d are the parameters of the chaotic system.

From the distribution maps of LEs of the 5D-HCCS and the 4D-CCS, we can see that although the general conservative chaotic systems have no attractor, they can only produce chaotic phenomenon, and their largest LE is always too small, thus their pseudo-randomness is not strong enough. On the contrary, the largest LE of the 5D-HCCS is large enough, thus it can fully function as a pseudo-random number generator required for chaotic image encryption. The 5D-HCCS not only has no attractor, but also produces hyperchaos, therefore we use it as a pseudorandom number generator in our image encryption system.

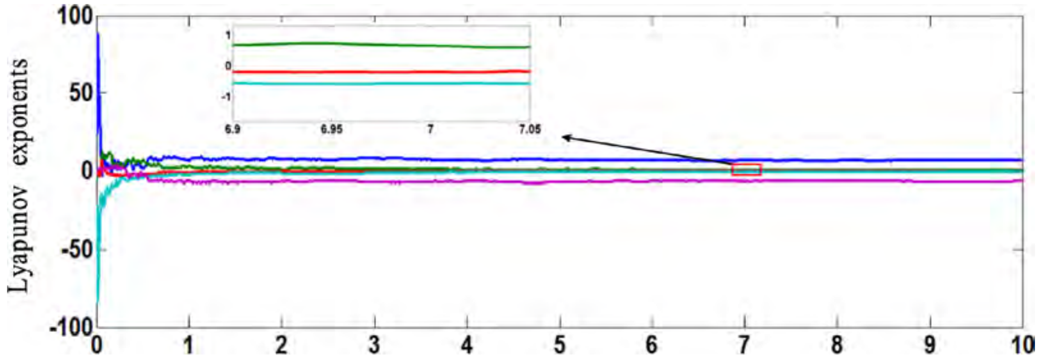


Fig. 1. (a) Finite-time local LEs of the 5D-HCCS with the initial value (*x*0, *y*0, *z*0, *u*0, *v*0) set as (0.25, 7, −2, 2.5, −0.7), and when $t \in (6.9, 7.05)$, the LEs stabilizes to(6.755464, 0.849765, 0.039421, −0.377339, −7.188469).

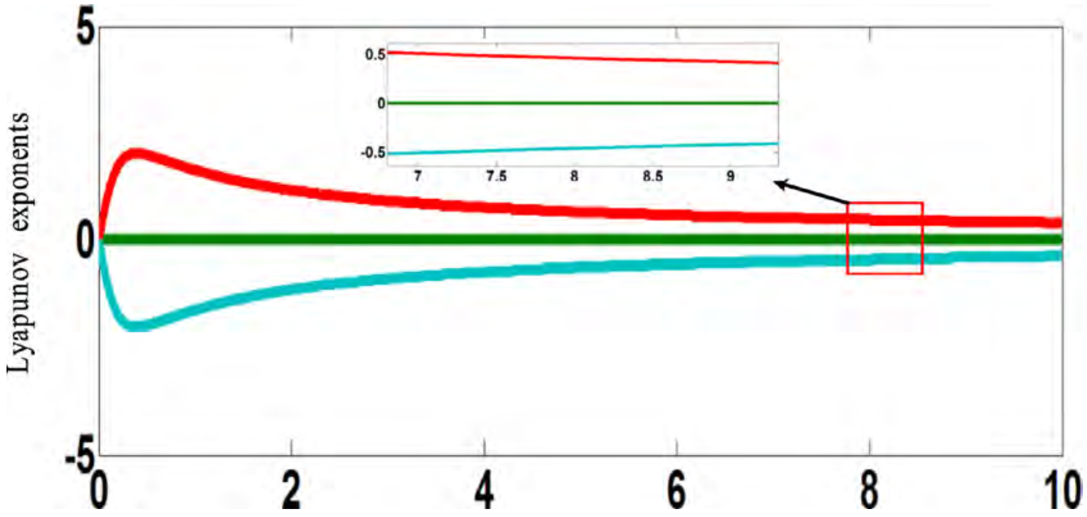


Fig. 2. Finite-time local LEs of the 4D-CCS with the initial value (*x*0, *y*0, *z*0, *w*0) set as (0.25, 7, −2, 2.5), when $t \in (7, 9)$, the LEs stabilizes to (-0.000000, −0.000000, 0.391222, −0.391222).

3. The proposed encryption scheme

3.1. Key generation process

In our encryption algorithm, the methods of key generating for scrambling and diffusion are different, and the specific ways are described in detail in this section.

3.1.1. Key generating for permutation process

A 256-bit external secret key is used in the permutation process, which is generated by SHA 256 hash function of the plain image. The 256-bit key K is divided into 8-bit blocks. It can be expressed as follows:

$$K = k1, k2, \ldots, k32, \text{subject to}: k_i = \{k_{i,0}, k_{i,1}, \ldots, k_{i,7}\} \tag{3}$$

where in the $k_{i,j}$, i represents the serial number of the blocks and j is the number of bits in the k_i. In order to make the 256-bit key grouping more clearly, here we give a schematic table of the key grouping, as shown in Table 1 below.

Obviously, the initial value (x_0^p, y_0^p, z_0^p, u_0^p, v_0^p) of chaotic system for scrambling process are generated by using these bit-level keys groups, and the formula is as follows.

$$\begin{cases} x_0^p = \frac{k_1+k_2+k_3+k_4+k_{29}+k_{30}+k_{31}+k_{32}-224}{8\times256} \\ y_0^p = \frac{k_5+k_6+k_7+k_8+k_{25}+k_{26}+k_{27}+k_{28}-244}{8\times256} + 1.5 \\ z_0^p = -\frac{k_9+k_{10}+k_{11}+k_{12}+k_{21}+k_{22}+k_{23}+k_{24}-236}{8\times256} \\ u_0^p = -\frac{k_{13}+k_{14}+k_{15}+k_{16}+k_{17}+k_{18}+k_{19}+k_{20}-216}{8\times256} - 1.5 \\ v_0^p = \frac{k_1+k_2+k_7+k_8+k_{23}+k_{24}+k_{19}+k_{20}-222}{8\times256} \end{cases} \tag{4}$$

According to the calculation formula given above, we know that the initial values of chaotic system used for permutation depend on the hash values of the original image. If there is a bit level difference between two original images, their hash values will be completely different and then chaotic sequences used for scrambling them will be very different. Therefore, our algorithm is highly sensitive to the plain image.

3.1.2. Key generating for diffusion process

The key for the diffusion process is determined by the size and pixel value of the plaintext. The size of the initial image is M × N, then according to the Eq. (5) we can calculate the initial values x_0^d, y_0^d, z_0^d, u_0^d, v_0^d of the chaotic system used for diffusing process.

Like the key in the scrambling phase, the key in the diffusion phase is directly related to the plaintext. As long as the size or

Table 1
The grouping of the secret key *K*.

K_1~K_4	K_5~K_8	K_9~K_{12}	K_{13}~K_{16}	K_{17}~K_{20}	K_{21}~K_{24}	K_{25}~K_{28}	K_{29}~K_{32}

pixel value of the original image changes slightly, the initial values of the chaotic sequence will change greatly. Due to the chaotic sequence is highly sensitive to the initial value, the chaotic sequence used to encrypt two slightly different images will be very different. As a result, our encryption algorithm can effectively resist the chosen-plaintext attacks.

$$\begin{cases} x_0^d = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(I_{i,j}+2)}{3\times M\times N}) \\ y_0^d = \bmod(\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(\frac{I_{i,j}+18}{9})}{3\times M\times N}\times 10^{10}, 10)+1 \\ z_0^d = -\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(\frac{I_{i,j}+36}{18})}{3\times M\times N} \\ u_0^d = \bmod(\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(\frac{I_{i,j}+188}{94})}{3\times M\times N}\times 10^{10}, -10) \\ v_0^d = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(\frac{I_{i,j}+228}{114})}{3\times M\times N} \end{cases} \tag{5}$$

where mod(*a*, *b*) represents the modular operation of *a* for *b* and $(\mathrm{a}+\mathrm{b})$ means bitand(a, b).

3.2. Closed-loop diffusion mechanism between blocks dependent on both plaintext and ciphertext (CDMBDPC)

The closed-loop diffusion mechanism between blocks dependent on both plaintext and ciphertext (CDMBDOC) is proposed in this paper to diffuse the scrambled image, and it can be denoted as Eq. (6).

$$C = \mathrm{CD}(Q, X2, Y2, Z2, U2, V2) \tag{6}$$

where Q represents the image block converted from the scrambled plaintext, CD(.) is the CDMBDPC function and *X*2, *Y*2, *Z*2, *U*2, *V*2 are five key blocks. The image block is composed of 4 × 4 small blocks, and the total number of blocks is M × N/(4 × 4) which is based on that the size of the scrambled plaintext as *M* × *N*. Firstly, we can obtain five sets of pseudorandom sequences by solving the equations of 5-dimensional conservative hyperchaotic system introduced in Section 2.1. Then we can convert them to 5 initial key blocks *X*2, *Y*2, *Z*2, *U*2, *V*2, which have the same size as *Q*.

The image is divided into small blocks based on the MATLAB cell array tool in this paper. Each block of an image *W* can be represented as *W*{*i, j*}, in which *i* represents the number of rows in the cell array, and *j* represents the number of columns. Therefore, in the order from left to right and from top to bottom, each block of an image *W* can be represented as $W\{1,1\}, W\{1,2\}, \dots, W\{1,r\}, W\{2,1\}, \dots, W\{r,r\}$, where *r* indicates the row and column labels of the last block and $r = \sqrt{(M\times N)/(4\times 4)}$. The diffusion of the current plaintext block *W*{*i, j*} is related to the previous ciphertext block *P*{*i, j*} and random ciphertext block *S*{*p, q*}, where the position of *P*{*i, j*} is located in front of the *W*{*i, j*} and the position of *S*{*p, q*} is located before *P*{*i, j*}. Since this is a block-by-block diffusion process, the index number *i* and *j* of the previous ciphertext block *P* depend on the index numbers of the current ciphertext block *C*{*i, j*}. Consequently, the specific *P*{*i, j*} can be given by the following formula:

$$P\{i,j\} = \begin{cases} C\{r,r\}, & i = 1\&\&j = 1 \\ C\{i-1,r\}, & i \neq 1\&\&j = 1 \\ C\{i,j-1\} & \text{others} \end{cases} \tag{7}$$

Meanwhile, the row number *p* and the column number *q* of the random ciphertext block *S* are also related to the index numbers of the current ciphertext block *C*{*i, j*} and they can be obtained according to Eqs. (8) and (9):

$$p = \begin{cases} 1, & i = 1 \\ \bmod(r\times r, i-1)+1 & i \neq 1 \end{cases} \tag{8}$$

$$q = \begin{cases} 1, & j = 1 \\ \bmod(r\times r, j-1)+1 & j \neq 1 \end{cases} \tag{9}$$

It can be seen from the above formulas that when *i* and *j* are both equal to 1, both *p* and *q* are set as 1. Fig. 3 gives a specific closed-loop diffusion flow chart. Assuming that the size of the scrambled image is *M* × *N* and the length of five sets of pseudorandom sequences are all *M* × *N*, then the detailed steps are as follows:

Step 1: First of all, the scrambled plaintext *Q* is evenly divided into small blocks, the size of which are all 4 × 4. Thereby, it can be represented as $Q\{1,1\}, Q\{1,2\}, \dots, Q\{1,r\}, Q\{2,1\}, \dots, Q\{r,r\}$. Meanwhile, five sets of pseudorandom sequences mentioned above are firstly transferred to the matrix of *M* × *N*, then we divide each one of them evenly into small blocks of 4 × 4 and they can be marked as base key blocks *X*2, *Y*2, *Z*2, *U*2, *V*2.

Step 2: Generate key blocks *b* and *d* for generating the initial first ciphertext block and the updated first ciphertext block. Since only one small block of *b* and *d* are involved in the generation of initial and updated first ciphertext blocks, the calculation method of the used blocks is given here.

$$b\{1,1\}(k,l) = \bmod(\mathrm{floor}(X2\{1,1\}(k,l)*10^{10}), 256) \tag{10}$$

$$d\{r,r\}(k,l) = \bmod(\mathrm{floor}(X2\{r,r\}(k,l)*10^{10}), 256) \tag{11}$$

where *k* and *l* represent the label of the row and the column in each block respectively. It is clearly that $1 \le k \le 4$, $1 \le l \le 4$ and in the following steps, they also represent these meanings.

Step 3: The initial first ciphertext block is constructed by the first plaintext block and the last plaintext block and the key block *b*. The calculation formula is as follows.

$$C\{1,1\}(k,l) = Q\{1,1\}(k,l) \oplus b\{1,1\}(k,l) \oplus Q\{r,r\}(k,l) \tag{12}$$

where *a*⊕*b* means bitxor(*a*, *b*) and it is also the same meaning when it appears in the formulas below.

Step 4: The key blocks *a* and c are gained by combining the initial key blocks *Y*2, *Z*2, *U*2, *V*2 with the previous ciphertext block *P*{*i, j*} according to Eqs. (13) and (14), in which the specific form of *P*{*i, j*} is given by Eq. (7).

$$a\{i,j\}(k,l) = \bmod(\mathrm{floor}((Y2\{i,j\}(k,l)+Z2\{i,j\}(k,l) +P\{i,j\}(k,l))*10^{8}), 256) \tag{13}$$

$$c\{i,j\}(k,l) = \bmod(\mathrm{floor}((U2\{i,j\}(k,l)+V2\{i,j\}(k,l) +P\{i,j\}(k,l))*10^{8}), 256) \tag{14}$$

where $1 \le i \le r$, $1 \le j \le r$, $1 \le k \le 4$, $1 \le l \le 4$, and the index numbers in the key blocks all correspond to the index number in the current plaintext block *Q* which need to be diffused.

Step 5: Set an intermediate variable *f*, which is determined by *P*{*i, j*}, and it can be calculated by the following formula:

$$f = \bmod(\mathrm{sum}(P\{i,j\}(k,l)*10^{14}), 2) \tag{15}$$

where the sum(.) operator means accumulating the values of every pixels in *P*. Obviously, The value of *f* is 0 or 1. We can select two encryption methods for the next small blocks to be diffused based on the value of *f*.

Step 6: The encryption of the second to the last plaintext block is performed according to the value of *f*. The specific encryption method of each ciphertext block is obtained by Eq. (16).

$$C\{i,j\}(k,l) = \begin{cases} Q\{i,j\}(k,l) \oplus a\{i,j\}(k,l) \oplus C\{p,q\}(k,l), & \text{if } f = 0, \\ Q\{i,j\}(k,l) \oplus c\{i,j\}(k,l) \oplus C\{p,q\}(k,l), & \text{if } f = 1. \end{cases} \tag{16}$$

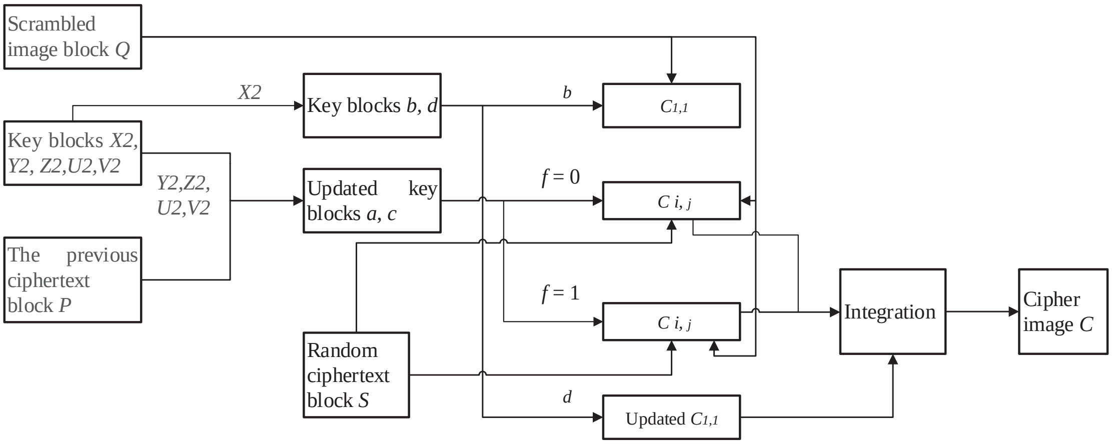

Scrambled image block Q
$X2$
Key blocks b, d
b
$C_{1,1}$
Key blocks $X2$, $Y2$, $Z2$,$U2$,$V2$
$Y2$,$Z2$, $U2$,$V2$
Updated key blocks a, c
$f = 0$
$C\ i, j$
The previous ciphertext block P
$f = 1$
$C\ i, j$
Integration
Cipher image C
Random ciphertext block S
d
Updated $C_{1,1}$

Fig. 3. The flow chart of CDMBDPC.

where the $C\{p, q\}(k, l)$ is the pixel of the ciphertext block S at any random position before the current plaintext block which need to be encrypted.

Step 7: Repeat **Step 6** until all plaintext blocks are encrypted.

Step 8: When the overall ciphertext is obtained, we need to use key block d, the original first ciphertext block and the last ciphertext block to update the first ciphertext block, so that the closed-loop diffusion can be realized. The updating way is as follows:

$$C'\{1, 1\}(k, l) = C\{1, 1\}(k, l) \oplus d\{r, r\}(k, l) \oplus C\{r, r\}(k, l) \tag{17}$$

where the $C\{1, 1\}(k, l)$ and $C'\{1, 1\}(k, l)$ are separately the original and the updated first ciphertext block.

Step 9: After all the ciphertext blocks are generated, we integrate the updated first ciphertext block and other ciphertext block to obtain a complete ciphertext block. Then we convert it to a matrix of the same size as the original plain image to obtain the final cipher image C.

The closed-loop diffusion mechanism between blocks dependent on both plaintext and ciphertext (CDMBDOC) explained in the steps above is given in Algorithm 1.

Algorithm 1 CDMBDOC.

Variables: t, the length of a small block; r, $r = \sqrt{(M \times N)/(4 \times 4)}$, the length of the cell array; p, q, the row number p and the column number q of the random ciphertext block S;
S, a random ciphertext block; P, the previous ciphertext block depended on the index number of the current ciphertext block $C\{i, j\}$; count, encryption rounds.
Input: Q, the image block converted from the scrambled plaintext, divided into small blocks, the size of which are all 4×4; X2, Y2, Z2, U2, V2, 5 initial key blocks.
Output: C, the cipher image.

```
cc=1;count=cc;
while(count)
  if(cc is equal to1)
    key blocks $b\{1,1\}(k,l) = \mathrm{mod}(\mathrm{floor}(X2\{1,1\}(k,l)*10^{10}), 256)$
    The first cipher block $C\{1,1\}(k,l) = Q\{1,1\}(k,l)$ bitxor $b\{1,1\}(k,l)$ bitxor $Q\{r,r\}(k,l)$
  else
    Q=C
  end if
  S=C{p,q}
  if(i and j are both equal to 1)
    P=C{r,r}
  elseif(i is not equal to 1 and j is equal to 1)
    P=C{i-1,r}
  else
    P=C{i,j-1}
  end if
  one key blocks $a\{i,j\}(k,l) = \mathrm{mod}(\mathrm{floor}((Y2\{i,j\}(k,l) + Z2\{i,j\}(k,l) + P\{i,j\}(k,l))*10^{8}), 256)$
  the other key blocks $c\{i,j\}(k,l) = \mathrm{mod}(\mathrm{floor}((U2\{i,j\}(k,l) + V2\{i,j\}(k,l) + P\{i,j\}(k,l))*10^{8}), 256)$
  $f = \mathrm{mod}(\mathrm{sum}(P\{i,j\}(k,l)*10^{14}), 2)$
  if(i is not equal to 1 or j is not equal to 1)
    if(f is equal to 0)
      $C\{i,j\}(k,l) = Q\{i,j\}(k,l)$ bitxor $a\{i,j\}(k,l)$ bitxor $C\{p,q\}(k,l)$
    elseif (f is equal to 1)
      $C\{i,j\}(k,l) = Q\{i,j\}(k,l)$ bitxor $c\{i,j\}(k,l)$ bitxor $C\{p,q\}(k,l)$
    end if
  end if
  key blocks $d\{r,r\}(k,l) = \mathrm{mod}(\mathrm{floor}(X2\{r,r\}(k,l)*10^{10}), 256)$
  The updated first cipher block $C\{1,1\}(k,l) = Q\{1,1\}(k,l)$ bitxor $b\{1,1\}(k,l)$ bitxor $Q\{r,r\}(k,l)$
count=count-1
end while
```

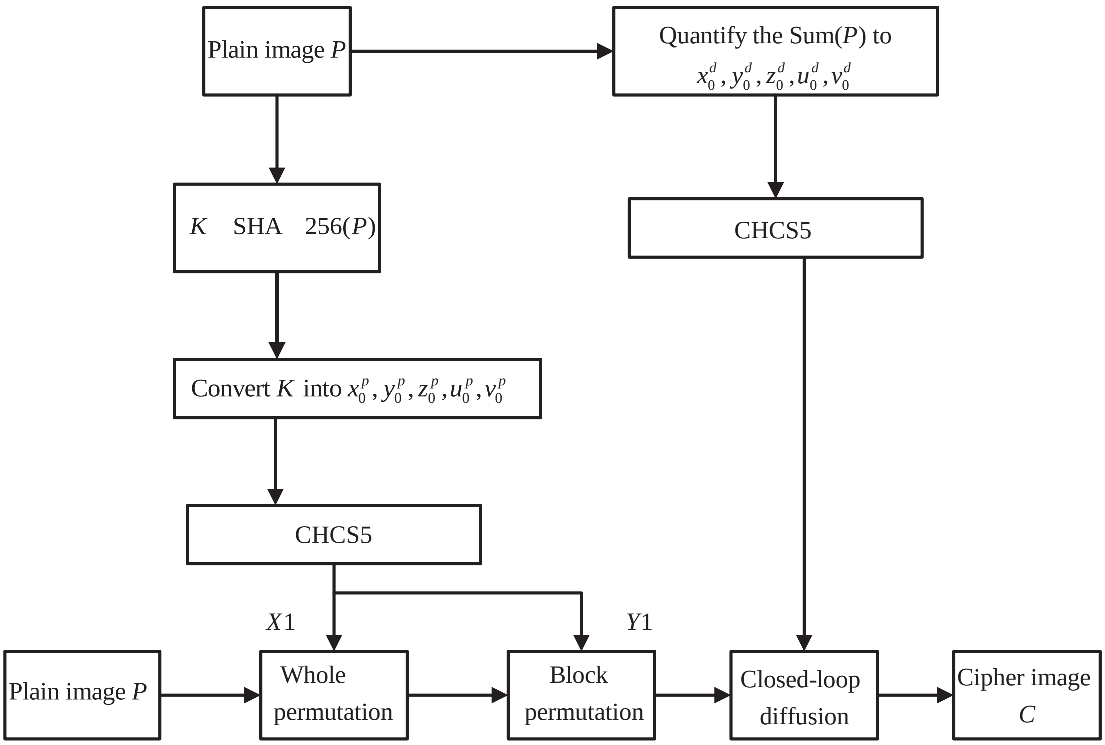


Fig. 4. The Flow chart of the encryption process.

From the perspective of the flow of the block-to-block closed-loop diffusion mechanism which is dependent on the plaintext and the ciphertext(CDMBDPC), we can see some highlights in our encryption scheme. First of all, the basic key block is formed by a certain number of iterations of the hyperchaotic system. In addition, the initial values of the chaotic system are generated by the quantification of the size and pixel values of the plaintext. While the key block used to encrypt each plaintext block is constructed by the base key block and the ciphertext block generated in front of the current plaintext block, which makes the key block determined by the plaintext and the ciphertext. Therefore, the key, plaintext and ciphertext constitute an interrelated organic whole, rather than being separated from each other, which better proves that the encryption system proposed in this paper cannot only resist the chosen-plaintext attack, but also resist the chosen-ciphertext attack, and thus has good security and reliability. Secondly, our diffusion process is a dynamic process. In addition to the key blocks which participate in each step of the encryption process is selected, a random ciphertext block at any position before the current plaintext block participates in the encryption process too. The dynamic structure realizes the true meaning of diffusion, that is, the change in the pixel values of any one of the encrypted image blocks is well used to affect the change in the pixel values of the unencrypted image blocks. Finally, by updating the first ciphertext block in **Step 8**, a closed-loop diffusion from $C\{1, 1\}$ to $C\{r, r\}$, and to $C'\{1, 1\}$ is achieved in a true sense, which is not reflected in other papers.

3.3. Encryption process

In our algorithm, the original image first experiences a global scrambling, and then experiences the block scrambling, and next through the closed-loop diffusion between blocks in Section 3.2. After these steps, we can get the cipher image.

Fig. 4 shows a complete schematic diagram of the encryption process and the plain image is denoted as P, whose size is L=$M \times N$, then the specific steps are described as below:

Step 1: Use a SHA-256 hash function for plain image P to produce a 256-bit key sequence K. Then use Eq. (4) to convert K into a set of initial value $(x_0^p, y_0^p, z_0^p, u_0^p, v_0^p)$ of the 5D conservative hyper-chaotic system(CHCS5), and the equation of CHCS5 is given by Eq. (1).

Step 2: We iterate the Eq. (1) $L+N0$ times using the initial value $(x_0^p, y_0^p, z_0^p, u_0^p, v_0^p)$ and the four parameters a, b, c, and d. The initial value is generated in the previous step and a, b, c, and d are set as 30, 30, 10 and 30 respectively. Then, we get 5 pseudorandom sequences. To get rid of the transient effect of the chaotic system, we discard the first $N0$ numbers of each sequence. Then five pseudorandom sequences $X1$, $Y1$, $Z1$, $U1$, $V1$ with L length can be obtained. Here, $N0$ is set to 1600.

Step 3: Convert the plain image matrix P into a sequence $P(i)$ of the same length as $X1$, and sort the $X1$ in ascending order, thereby obtaining an index sequence $T = \{t(i)\}_1^L$, where $t(i)$ means the component of T.

Step 4: Scramble the plain image sequence $P(i)$ to get the shuffled sequence $P'(i)$ by

$$P'(i) = P(t(i)) \tag{18}$$

where $i = 1, 2, ..., L$.

Step 5: Convert the image after the overall scrambling into a matrix of $M \times N$, and then evenly divide it into 4 small blocks and the number of blocks is $r \times r = M \times N/(4 \times 4)$. Then the image can be represented as $P'\{1,1\}, P'\{1,2\}, \ldots, P'\{1,r\}, P'\{2,1\}, \ldots, P'\{r,r\}$ or in the sense of using a one-dimensional sequence to represent each block, the image can be expressed as $P'(1), P'(2), \ldots, P'(r \times r)$. Here, we can set the index numbers of the blocks as $G(i) = 1, 2, ..., r \times r$, where $i =$

$1, 2, \ldots, r \times r$. Thereby, the overall-scrambled image block can be represented as $P'(G(i)$.

Step 6: Extract $r \times r$ numbers from $X1$ to form a new chaotic sequence $X1'$, and then sort $X1'$ in ascending order to get an index sequence $T' = \{t'(i)\}_1^{r\times r}$, where $t'(i)$ means the component of T'.

Step 7: Scramble $G(i)$ to get the new index numbers of blocks $G'(i)$ by

$$G'(i) = G(t'(i)) \tag{19}$$

where $i = 1, 2, \ldots, r \times r$.

Step 8: Scramble the overall-scrambled image block $P'(G(i))$ to get the shuffled image block $Q(G'(i))$ by

$$Q(G'(i)) = P'(G(t'(i))) \tag{20}$$

After the block scrambling, we get a new scrambled image Q which can be expresesed as $Q\{1,1\}, Q\{1,2\}, \ldots, Q\{1,r\}, Q\{2,1\}, \ldots, Q\{r,r\}$.

Step 9: After two scrambling, the permutated image Q can be sent to diffusion stage. Firstly, we calculate the Eq. (5) based on the size of the plain image and the sum of each of its pixel values to obtain the five keys of the diffusion phase, namely the other set of initial value (x_0^d, y_0^d, z_0^d, u_0^d, v_0^d) of the CHCS5.

Step 10: We iterate the Eq. (1) $L + N1$ times with the initial value x_0^d, y_0^d, z_0^d, u_0^d, v_0^d and the four parameters *a, b, c*, and *d*. The initial value is generated in the previous step and sent to the CHCS5, and *a, b, c*, and *d* are set as 30, 30, 10 and 30 respectively. Then we can get 5 pseudorandom sequences. To get rid of the transient effect of the chaotic system, we discard the first $N1$ numbers of each sequence. Then five pseudorandom sequences $X2$, $Y2$, $Z2$, $U2$, $V2$ with L length can be obtained. Here, $N1$ is set to 1500.

Step 11: Perform the first to the ninth step in Section 3.2 in order to complete the closed-loop diffusion between blocks dependent on both plaintext and ciphertext (CDMBDPC).

Since the scrambling and diffusion phases are successively experienced, we can get a complete cipher image C.

3.4. Decryption process

The decryption scheme is the reverse process of the encryption scheme. The key used for decryption is exactly the same as the encryption, and the key should be transmitted by the sender to the receiver via a secure transmission channel before decryption. In the algorithm proposed in this paper, the secret key consists of 10 values, namely the first set of initial value of chaotic sequences used in the scrambling phase ($x_0^p, y_0^p, z_0^p, u_0^p, v_0^p$) and the other set of initial value of chaotic sequences used in the diffusion phase (x_0^d, y_0^d, z_0^d, u_0^d, v_0^d).

4. Simulation results and performance analysis

This part gives detailed security analyses and experimental results to evaluate the performance of the proposed algorithm. To verify the validity and efficiency of our proposed algorithm, several numerical simulations performed on several images taken from a well-known database [43] are conducted on the Matlab2016a platform and discussed in the following subsections. The chosen sample image are the standard 256 grayscale images of Black, White, Peppers, Elaine, Lena, Goldhill, Baboon, Man, Airplane, and Airport, where the first four images, the middle three images and the last three images measure 256 × 256 pixels, 512 × 512 pixels and 1024 × 1024 pixels respectively. Fig. 5 shows the encryption and decryption results for the Elaine, Baboon, and Airport images. According to Fig. 5, we can conclude that the encrypted images were similar to the noisy images without any visual information leakage, and the decrypted images with the corrected key were identical to the plain images. In addition, in the following performance analysis, the performance parameters of our encryption scheme are obtained after only one round of encryption.

4.1. Key space analysis

The size of the key space is the total number of different keys that can be used in a cryptosystem. A larger key space means that the algorithm has higher security. To guarantee the security of the cryptosystem, the key space should be larger than 2^{112} to resist brute-force attacks [44]. In the proposed algorithm, the key comprises the initial values x_0^p, y_0^p, z_0^p, u_0^p, v_0^p, x_0^d, y_0^d, z_0^d, u_0^d, and v_0^d. We assume that the initial values are double-precision numbers. Because the computational precision of the double-precision numbers is 10^{-16}, the size of the key space of the proposed algorithm for one round of encryption would be bigger than

$$[(10^{16})^3 \times (0.49 \times 10^{16})^2]^2 \approx 2^{399},$$

which is sufficiently large to resist all types of brute-force attacks.

4.2. Key sensitivity analysis

A high security cryptosystem must be very sensitive to tiny differences in secret keys. In the encryption process, almost completely different cipher images should be produced when slightly different keys are employed to encrypt the same plain image. Similarly, a slight change to the key used for decryption will lead to an unsuccessful decryption. In this test, we check the sensitivity of the key in the encryption phase and decryption phase.

In the encryption phase, the original keys and the slightly modified keys are used to encrypt the Elaine (256*256) image respectively. The original keys are set as ($x_0^p = 0.498046875000000$, $y_0^p = 1.934082031250000$, $z_0^p = -0.294921875000000$, $u_0^p = -2.110839843750000$, $v_0^p = 0.300292968750000$, $x_0^d = 0.330312093098958$, $y_0^d = 7.822916984558106$, $z_0^d = -0.335266113281250$, $u_0^d = -0.692708015441895$, $v_0^d = 0.356084187825521$) and the slightly modified keys are set as ($x_0^p = 0.498046875000001$, $y_0^p = 1.934082031250000$, $z_0^p = -0.294921875000000$, $u_0^p = -2.110839843750000$, $v_0^p = 0.300292968750000$, $x_0^d = 0.330312093098958$, $y_0^d = 7.822916984558106$, $z_0^d = -0.335266113281250$, $u_0^d = -0.692708015441895$, $v_0^d = 0.356084187825521$). The original Elaine image is shown in Fig. 6(a), and the two cipher images are shown in Fig. 6(b) and (c) which correspond to original keys and slightly modified keys respectively. The difference between the two cipher images is shown in Fig. 6(d). It is clear that slightly different keys will produce two completely different cipher images.

In the decryption phase, the correct and the incorrect decryption keys are used to decrypt the same cipher image respectively. The correct and incorrect decryption keys are the same as the original and the slightly modified encryption keys respectively. The original Elaine image and the corresponding cipher image of the original keys are shown in Fig. 7(a) and (b). The decrypted images using the incorrect decryption keys and the correct decryption keys are shown in Fig. 7 (c) and (d).

4.3. Histogram analysis

As we know, the image histogram represents the distribution of the pixel intensity values in an image. The more random are

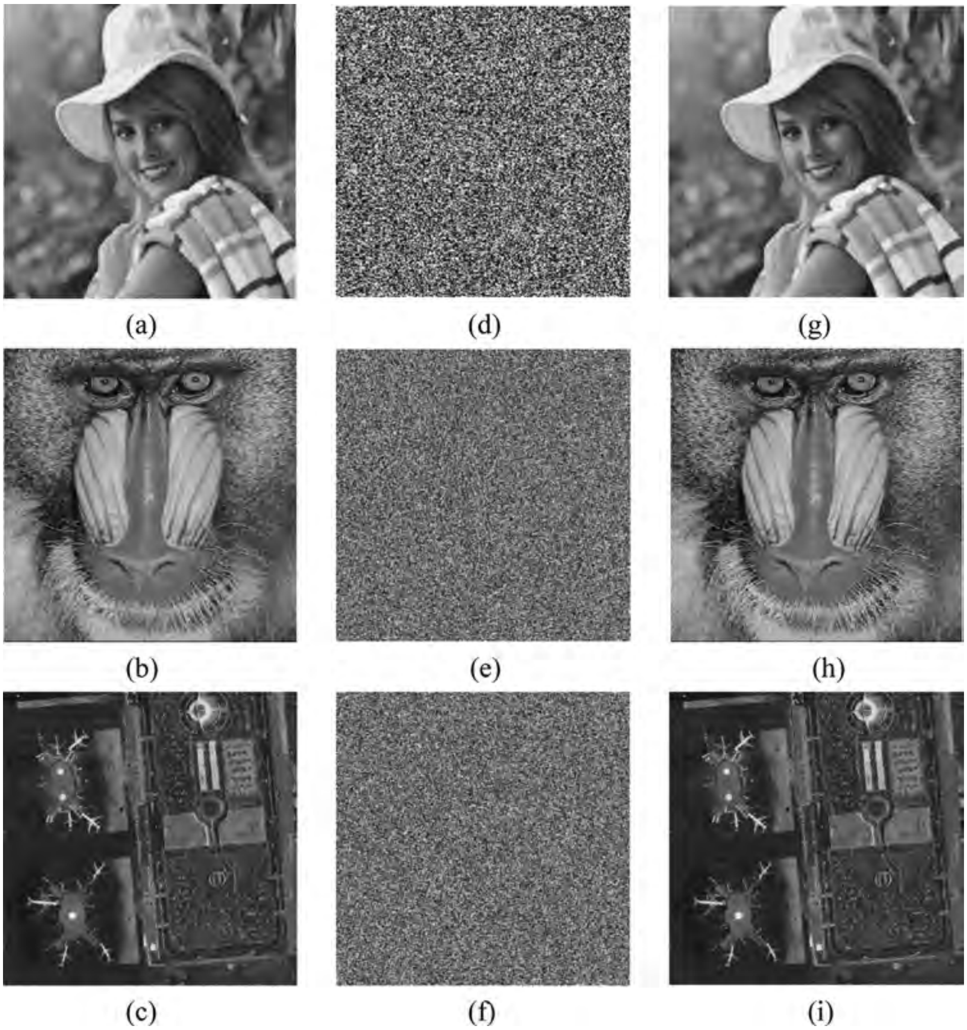


Fig. 5. Experimental results: (a)–(c) plain Elaine, Baboon, and Airport images, respectively; (d)–(f) cipher images for (a)–(c); and (j)–(l) images retrieved from (d)–(f) with the corrected key.

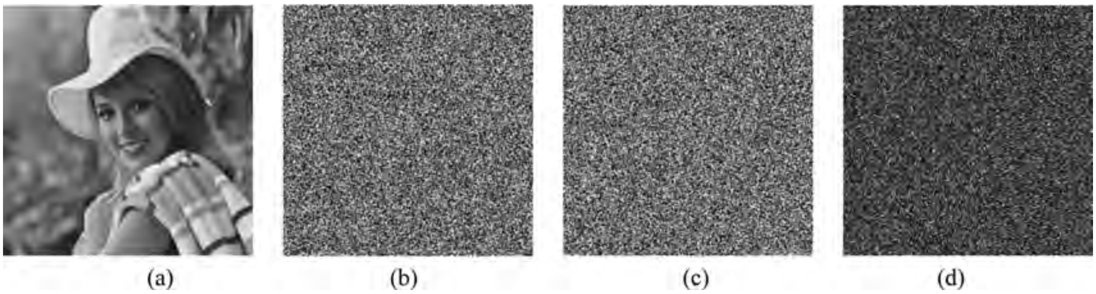


Fig. 6. Key sensitivity test for image encryption: (a) Elaine image, (b) encrypted image using the original key, (c) encrypted image using the modified key and (d) the difference between (b) and (c).

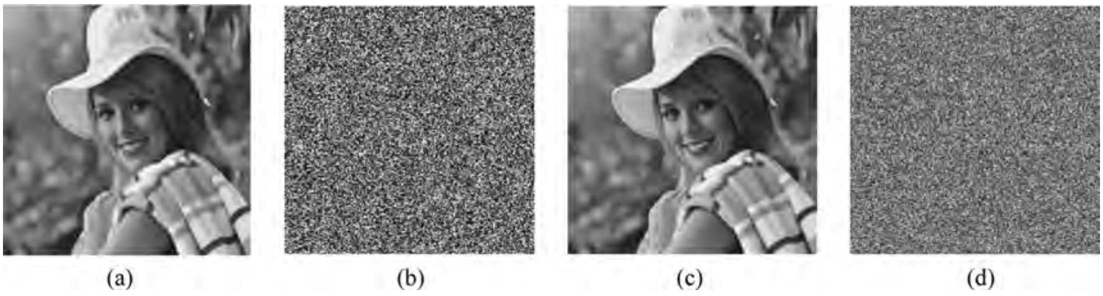


Fig. 7. Key sensitivity test for image decryption: (a) Elaine image, (b) encrypted image using the original key, (c) decrypted image using the correct decryption key and (d) decrypted image using the incorrect decryption key.

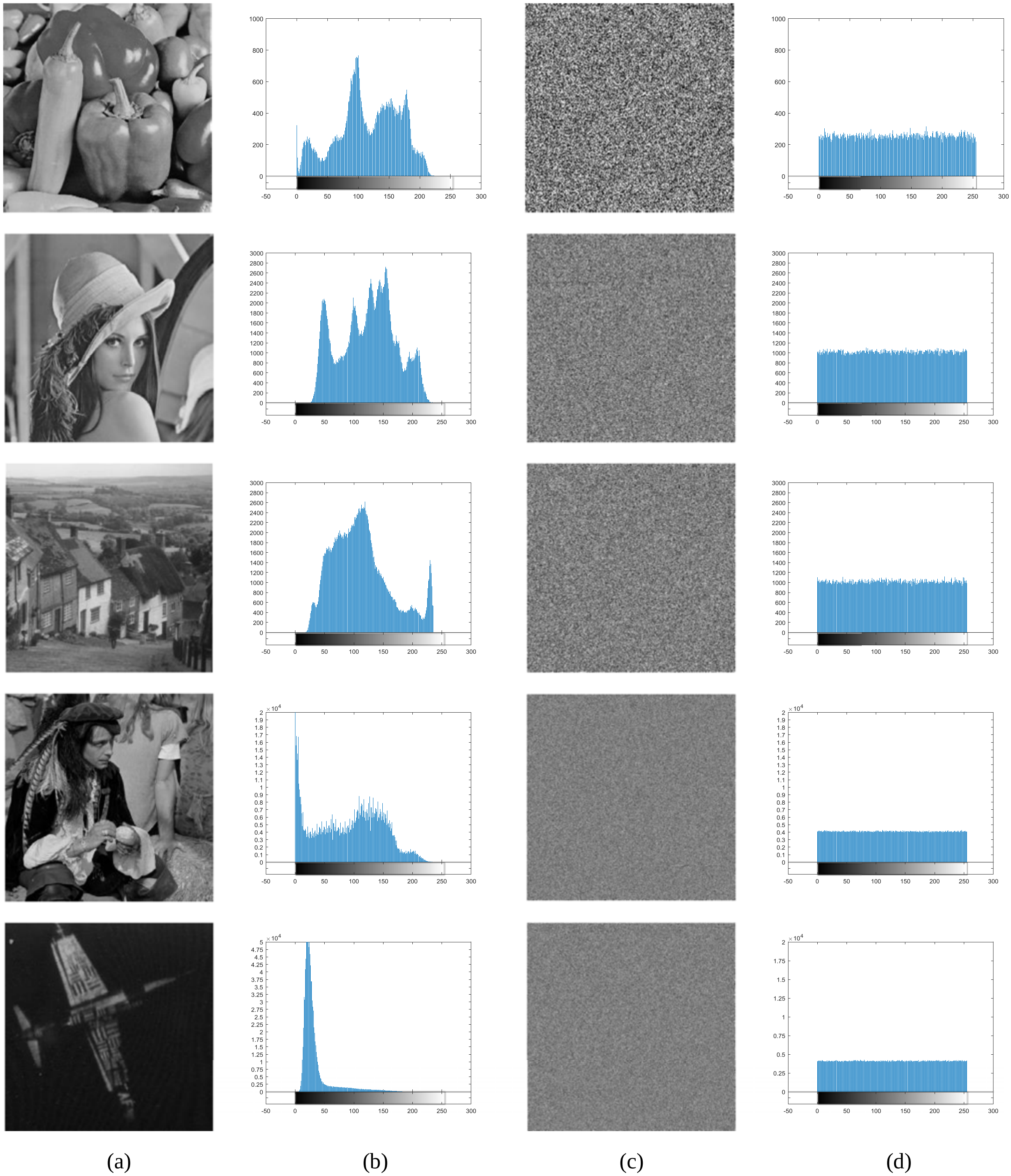


Fig. 8. Histogram plots of several images: Histograms (b) of the original images (a); histograms (d) of the related cipher images (c).

the pixel values of the cipher image and the better is the performance against statistical attacks, the more uniform and flattened will be the distribution of the histogram of the cypher image. Fig. 8 presents original test images (a), their cipher image counterparts (c) and histograms plots of them (b, d). As seen from the figure, after one round encryption process, all the test images with non-uniform histogram distributions are changed to cipher images with uniformly distributed histograms. Thus it is also proved that the proposed image encryption algorithm can resist statistical attacks well.

4.4. Correlation analysis

There is a high correlation between the adjacent pixel values of plain images in horizontal, vertical and diagonal directions. An ideal encryption algorithm can greatly reduce the correlation of

Table 2
Correlation coefficients between adjacent pixels in the plain and cipher images.

Test images	Original image			Cipher image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Peppers	0.868501	0.933391	0.892350	0.000476	−0.009531	0.007338
Elaine	0.955971	0.936554	0.873894	−0.012954	0.006684	0.033789
Lena	0.964227	0.982430	0.965609	−0.038118	−0.029142	0.002736
Goldhill	0.960441	0.961766	0.846811	0.005502	0.001494	−0.020027
Baboon	0.554351	0.755668	0.586725	−0.014340	0.011214	0.001345
Man	0.987193	0.992968	0.946781	−0.009565	−0.000507	0.007117
Airplane	0.824170	0.566200	0.214058	−0.008918	−0.015649	−0.002008
Airport	0.883983	0.660332	0.241429	0.004946	0.017340	−0.006176

adjacent pixels in the three directions of the cipher image to resist statistical attacks. To test the correlations between two adjacent pixels in an image, we randomly select N pairs of two adjacent pixels (either horizontal, vertical, or diagonal) from the plain image and its corresponding cipher image and calculate the correlation coefficient r_{xy} for each pair using the following formula:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{21}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2 \tag{22}$$

$$\mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)) \tag{23}$$

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x) \bullet D(y)}} \tag{24}$$

where we used $N = 10000$ and x_i and y_i($i = 1, 2, \ldots, N$) are the grayscale values of two adjacent pixels in the image. *E(x)* and *D(x)* denote the expectation and variance of variable x, respectively. Fig. 9 plots the correlation of two adjacent pixels of the plain image Lena and its cipher image in the horizontal, vertical and diagonal direction. Moreover, Table 2 shows the coefficients of two adjacent pixels in the plain and cipher images. According to the Fig. 9, it is clear that the distributions of adjacent pixels in the original image are highly concentrated, which means that the original image has a strong correlation. However, the distributions of the adjacent pixels in ciphered image are random, which means that the cipher image has a low correlation. From the Table 2, we can see that the correlation coefficients of two adjacent pixels in the original images are all bigger than 0.5, so the correlations between adjacent pixels are strong. Nevertheless, the correlation coefficients of the cipher images are all smaller than 0.1, which indicates the strong correlations between adjacent pixels in the plain images are greatly reduced in the cipher images.

4.5. Entropy analysis

Normally, we use information entropy to characterize the intensity of randomness of a system. For the encryption system, the closer to 8 the measured information entropy of the cipher image is, the more secure the encryption algorithm is, and the more difficult the attacker can decode the encryption algorithm. For a gray-scale image, the information entropy measures the distribution of intensity values in it. Information entropy with a greater value means more uniform distribution of intensity values [45]. The entropy is defined as

$$H(m) = \sum_{i=0}^{M-1} p(m) \log \frac{1}{p(m_i)} \tag{25}$$

where M is the total number of symbols $m_i \in m$ and $p(m_i)$ denotes the probability of symbols. We can get theoretical value $H(m) = 8$ by calculating Eq. (25). Therefore, the more the information entropy gets close to 8, the less possible for attackers to decode cipher images. Table 3 shows the measurement results of information entropy and the comparison with previous studies [13,24,30]. It should be noted here that just one round encryption are performed to achieve the measured information entropy value and the algorithms, and the compared other encryption schemes also made one round encryption to achieve their information entropy values, thus this comparison is feasible. From Table 3, it can be known that entropies of cipher images are close to 8, so the proposed algorithm has a good property of information entropy. In addition, compared with the previous studies, it can be seen that our information entropy results have made some progress in some ways.

4.6. Differential attack analysis

In order to resist the differential attack, a good encryption system should be highly sensitive to subtle changes in the plain image. Generally, NPCR (Number of Pixels Change Rate) and UACI (Unified Average Changing Intensity) are used to measure invulnerability to differential attacks [45–47] and they are calculated as follows:

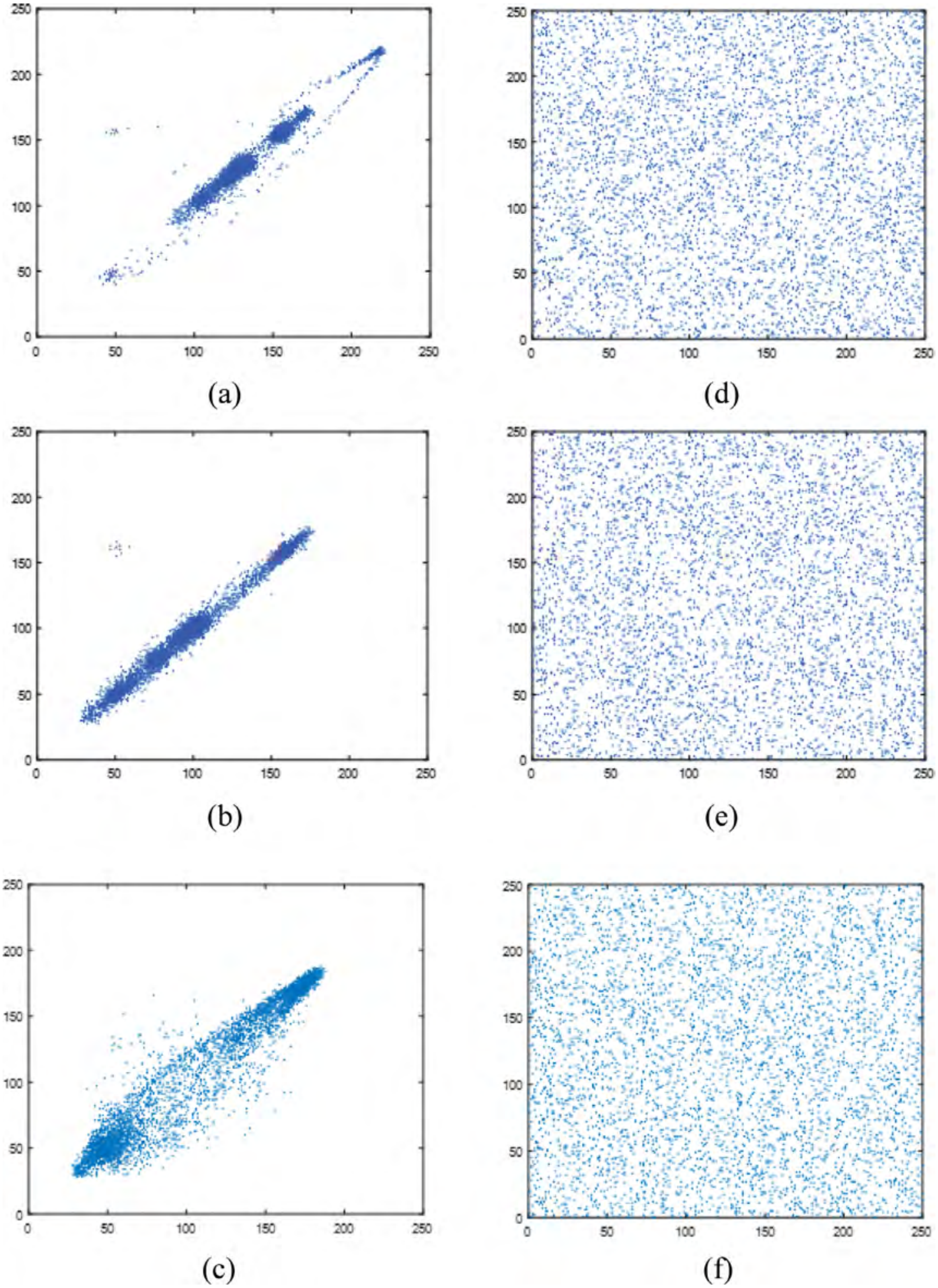
$$\mathrm{NPCR} = \frac{1}{W \times H}\sum_{i-1}^{W}\sum_{j=1}^{H} D(i,j) \times 100\% \tag{26}$$

$$\mathrm{UACI} = \frac{1}{W \times H}\sum_{i=1}^{W}\sum_{j=1}^{H} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\% \tag{27}$$

where C_1 and C_2 are the two cipher images encrypted by the proposed algorithm with the same size $W \times H$ in consequence of a slight change in the chosen plain image and $D(i, j)$ is defined as:

$$D(i,j) = \begin{cases} 0 & C_1(i,j) = C_2(i,j) \\ 1 & C_1(i,j) \neq C_2(i,j) \end{cases} \tag{28}$$

In order to test the sensitivity of the plain images, one pixel is randomly selected from each plain image. Then we alter the last bit of the pixel at the same location to obtain a modified plain image. The original image and the slightly modified image are encrypted with the same key, and two cipher images are generated. The NPCR and UACI values between the two encrypted images are calculated. For each sample image, we make 30 sets of tests with only one pixel variation and calculate the average values of UACI and NPCR, whose results are shown in Table 4. In addition, we select the position of the changed pixel at the front, middle and the end of the three groups of image respectively to compute the values of UACI and NPCR. The test results are presented in Table 5 and it proves that our test is valid.



(a)

(d)

(b)

(e)

(c)

(f)

Fig. 9. Correlation coefficients computed for the image Lena. (a) horizontal correlation of original image; (b) vertical correlation of original image; (c) diagonal correlation of original image; (d) horizontal correlation of cipher image; (e) vertical correlation of cipher image; (f) diagonal correlation of original image.

From the two tables, it can be seen that the value of NPCR is close to 99.6% and the value of UACI is close to 33.4%. Therefore, the proposed algorithm can effectively resist differential attacks.

4.7. Plaintext and ciphertext attacks analysis

As we all know, a good encryption scheme needs to have good performance against known-plaintext attacks and chosen-plaintext attacks. Some encryption algorithms that are not very resistant to known plaintext attacks and chosen-plaintext attacks can be easily deciphered, making the encryption less secure. For example, Ref. [48] and Ref. [49] used the proposed chosen-plaintext attack algorithms to decipher the existing image encryption schemes and gave two improved encryption schemes. In our image encryption process, there are some ways to enhance the performance of our encryption scheme to resist known-plaintext attacks and chosen-

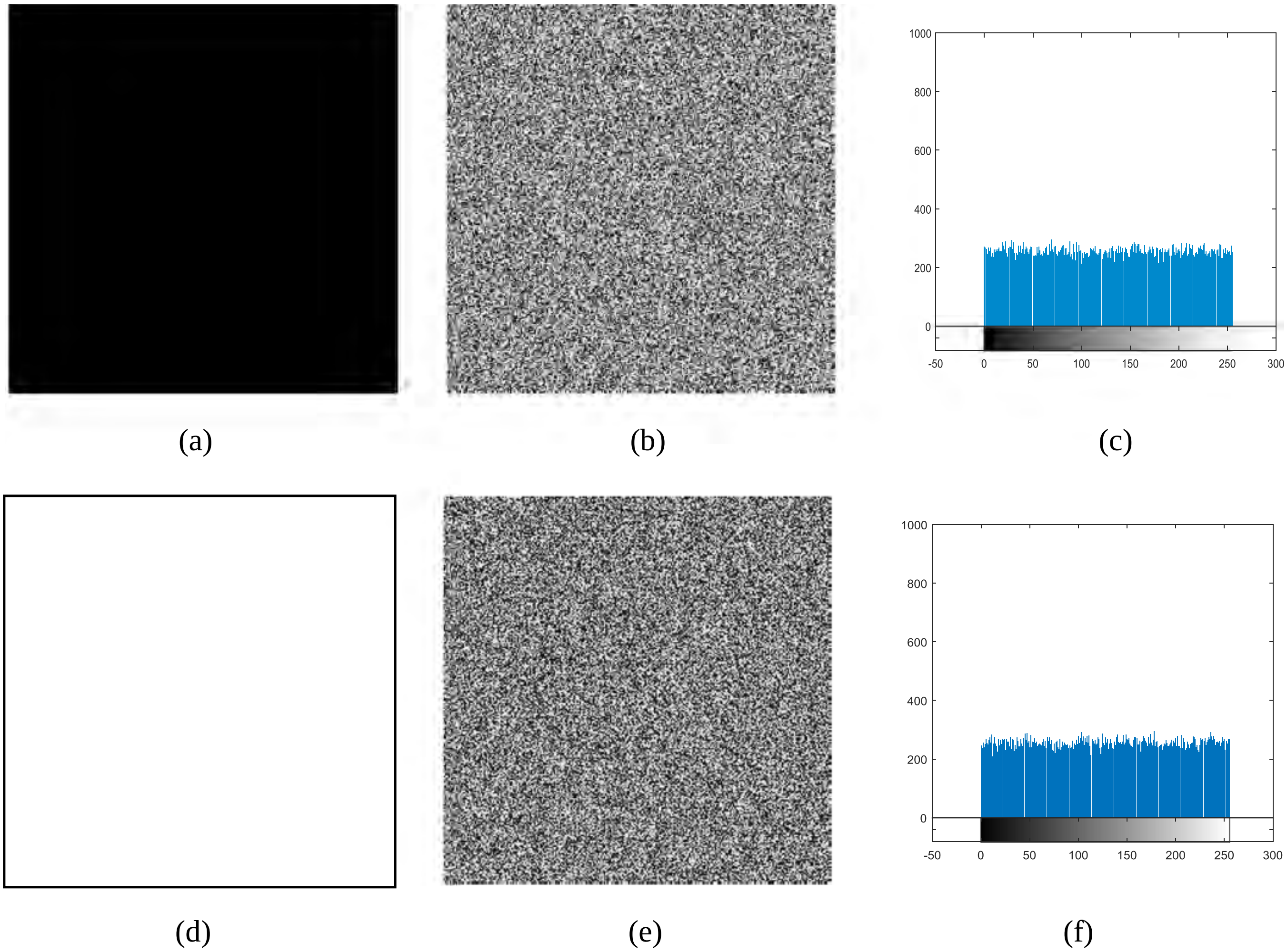


Fig. 10. Experimental results for the special images: (a) all-black original image; (b) the cipher image of (a); (c) the histogram of(b); (d) all-white original image; (e) the cipher image of (d); (f) the histogram of(e).

Table 3
The measurement results of information entropy.

Test image	size	Entropy of original image	Entropy of cipher image	Ref. [13]	Ref. [24]	Ref. [30]
Black	256*256	0	7.997536	–	–	–
peppers	256*256	7.539904	7.997002	–	7.9973	–
Elaine	256*256	7.509576	7.997222	–	–	–
Lena	512*512	7.445507	7.999239	7.9993	7.9971	7.999276
Baboon	512*512	7.357949	7.999301	7.9993	7.9992	–
Goldhill	512*512	7.477780	7.999183	7.9994	–	–
Airport	1024*1024	6.830330	7.999818	–	–	7.999832
Airplane	1024*1024	5.641454	7.999859	–	–	–
Man	1024*1024	7.523737	7.999829	–	–	7.999826

Table 4
Average values of NPCR and UACI for different images.

Test images	Elaine	Peppers	Lena	Baboon	Man	Airport
UACI (%)	33.4333	33.4245	33.4523	33.4039	33.4713	33.2875
NPCR (%)	99.6022	99.6115	99.6114	99.6016	99.6089	99.6043

plaintext attacks. Firstly, the key stream for scrambling is generated by performing a hash function on the pixel values of the plain image, and the key stream for diffusion is generated by performing a series of quantization on the sum of the pixel values of the plaintext image. It can be seen that the generation process of key stream in our encryption scheme is closely related to the original image, thus our encryption scheme is highly sensitive to the change of the plain image. Secondly, in the diffusion process, the key block, the plain image block, and the cipher image block form an organic whole. Moreover, the pixel values of the current plaintext block and the generated ciphertext blocks can affect the generation process of the next ciphertext image block. Therefore, our encryption scheme cannot only resist known-plaintext and chosen-plaintext attacks, but also can resist known ciphertext attacks.

Table 5
The three groups of image for different positions.

Test images		(1,2)	(125,127)	(250,253)
Elaine	UACI (%)	33.6424	33.3921	33.4139
	NPCR (%)	99.6384	99.5956	99.6078
Lena	UACI (%)	33.4099	33.4807	33.4809
	NPCR (%)	99.6315	99.6124	99.5876
Baboon	UACI (%)	33.4024	32.9595	33.3699
	NPCR (%)	99.6117	99.5407	99.6071

Table 6
Encryption results of all-black and all-white images.

Images	NPCR (%)	UNCI (%)	Information entropy	Correlation coefficient		
				Horizonal	Vertical	Diagonal
Cipher image for all-black	99.6328	33.4965	7.997536	0.009463	−0.024885	0.002679
Cipher image for all-white	99.6129	33.5079	7.997426	0.031301	0.021430	0.008299

Some attackers may choose plain images which are all-black or all-white to decipher the encryption scheme. When these two special images are encrypted, the scrambling procedure will not work, then the overall performance of encryption will be worse in some way. In this two special encryption process, the attacker may find some information missing from the encryption process to decipher the encryption scheme. However, in our encryption scheme, in addition to the position of image pixels can be changed by the scrambling process, the size of pixel values can also be changed by the closed-loop diffusion process, thus this encryption method can ensure high security whatever original image is. Fig. 10 below shows the all-black and all-white original image, their cipher images and the histograms of the cipher images. All of the images measured 256 × 256. Table 6 shows the information entropy, NPCR, UNCI, and correlation coefficients of the encryption process of these two special images.

From Fig. 10, we can see that the distribution of the histograms of the cipher images of all-black and all-white images are both uniform, thus the decipherer cannot obtain any useful information from the cipher images and the encryption system cannot be broken. Therefore, it confirms that our image encryption algorithm can resist the chosen-plaintext attacks. In addition, from the test data listed in Table 6, we can see the encryption processes of these two special images both are quite safe and highly sensitive to the original images.

4.8. Peak signal-to- noise ratio analysis

We use the PSNR (peak signal-to-noise ratio) to measure encryption performance and it can be calculated by the following equations:

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} [P(i,j) - C(i,j)]^2 \tag{29}$$

$$\mathrm{PSNR} = 10 \times \lg\left(\frac{I_{\max}^2}{\mathrm{MSE}}\right) \tag{30}$$

where $M \times N$ is the size of image. Moreover, $P(i, j)$ and $C(i, j)$ are the pixel values of original image and encrypted image respectively, and $I_{\max}$ is the maximum pixel value of the image. It is obvious that the value of PSNR should be as small as possible to ensure the efficiency of the algorithm.

To test the quality of the encryption scheme, we test the value of PSNR when encrypting the image Peppers, Lena, Brain, Baboon and the results and comparative analysis with previous research data are presented in the Table 7 below. From the Table 7, we can see that our measured PSNR values for different encrypted images are all very low and when encrypting Lena, Peppers and Brain, the proposed algorithm presents smaller PSNR values than those in Ref. [50,51]. Therefore, the encryption quality of the proposed algorithm is superior.

Table 7
PSNR for the encryption.

PSNR	Ours	Ref. [50]	Ref. [51]
Peppers(256*256)	9.0178	9.0442	8.8772
Lena(256*256)	8.4265	9.1772	9.2337
Brain(256*256)	5.7800	8.9604	9.1145
Baboon(512*512)	9.5322	–	–

5. Conclusion

In this paper, a novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks is proposed. We use a five-dimensional conservative hyperchaotic system to provide pseudo-random sequences in the scrambling and diffusion stages, and the methods of generating the initial values of the chaotic sequences of the two stages are different. The first group is obtained by applying a hash function to the plaintext image, and the second group is obtained by quantizing the pixel values of the plaintext image. First of all, our encryption scheme has superiority to other schemes which are based on general chaotic systems because that the conservative hyperchaotic system we used not only has good randomness and ergodicity, but also has the anti-reconstruction attack characteristic. Secondly, our encryption process focuses on the innovation of the diffusion structure, which is different from the most existing studies that focus on the innovation of scrambling structure. This specific encryption process can be briefly described as: scrambling the image as a whole, scrambling between blocks, and inter-block closed-loop diffusion performed on the scrambled image. The key block used for generating each cipher block is determined by the cipher block located in front of the current cipher block. In addition, a random cipher block before the current cipher block is also involved in the generation of the current cipher block. Besides, the pixel value of the first cipher block is updated after the last cipher block is generated. Therefore, the key block depends on both plaintext and ciphertext thus to ensure the sensitivity and security of the encryption system. Moreover, a large number of theoretical analysis and performance tests have been tested and evaluated for our encryption scheme. Obviously, from the results we can see that the encryption scheme proposed in this paper has strong reliability and security.

Declaration of Competing Interest

None.

CRediT authorship contribution statement

Minjun Zhou: Conceptualization, Methodology, Software, Validation, Data curation, Formal analysis, Writing - original draft, Visualization, Investigation, Writing - review & editing. **Chunhua Wang:** Resources, Supervision, Writing - review & editing, Project administration, Funding acquisition.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61971185), the Major Research Plan of the National Natural Science Foundation of China (No. 91964108) and the Open Fund Project of Key Laboratory in Hunan Universities (No. 18K010).

References

[1] C. Li, Y. Zhang, E.Y. Xie, When an attacker meets a cipher-image in 2018: a year in review, J. Inf. Secur. Appl. 48 (2019) 102361.
[2] C. Chen, K. Sun, S. He, An improved image encryption algorithm with finite computing precision, Signal Process. (2019) 107340.
[3] V.M. Silva-García, R. Flores-Carapia, C. Rentería-Márquez, et al., Substitution box generation using Chaos: an image encryption application, Appl. Math. Comput. 332 (2018) 123–135.
[4] H. Tang, Q.T. Sun, X. Yang, et al., A network coding and DES based dynamic encryption scheme for moving target defense, IEEE Access 6 (2018) 26059–26068.
[5] H. Lin, C. Wang, Influences of electromagnetic radiation distribution on chaotic dynamics of a neural network, Appl. Math. Comput. 369 (2020) 124840.
[6] Q. Zhao, C. Wang, X. Zhang, A universal emulator for memristor, memcapacitor, and meminductor and its chaotic circuit, Chaos Interdiscip. J. Nonlinear Sci. 29 (1) (2019) 013141.
[7] X. Zhang, C. Wang, Multiscroll hyperchaotic system with hidden attractors and its circuit implementation, Int. J. Bifurc. Chaos 29 (09) (2019) 1950117.
[8] X. Zhang, C. Wang, W. Yao, et al., Chaotic system with bondorbital attractors, Nonlinear Dyn. 97 (4) (2019) 2159–2174.
[9] Q. Deng, C. Wang, Multi-scroll hidden attractors with two stable equilibrium points, Chaos Interdiscip. J. Nonlinear Sci. 29 (9) (2019) 093112.
[10] F. Yu, L. Liu, B. He, et al., Analysis and FPGA realization of a novel 5D hyperchaotic four-wing memristive system, active control synchronization, and secure communication application, Complexity 2019 (2019) 2019.
[11] J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps, Int. J. Bifurc. Chaos 8 (06) (1998) 1259–1284.
[12] Y.Q. Zhang, X.Y. Wang, J. Liu, et al., An image encryption scheme based on the MLNCML system using DNA sequences, Opt. Lasers Eng. 82 (2016) 95–103.
[13] X. Chai, Y. Chen, L. Broyde, A novel chaos-based image encryption algorithm using DNA sequence operations, Opt. Lasers Eng. 88 (2017) 197–213.
[14] X. Chai, X. Fu, Z. Gan, et al., A color image cryptosystem based on dynamic DNA encryption and chaos, Signal Process. 155 (2019) 44–62.
[15] J. Wu, X. Liao, B. Yang, Image encryption using 2D Hénon-Sine map and DNA approach, Signal Process. 153 (2018) 11–23.
[16] X. Zhang, Z. Zhou, Y. Niu, An image encryption method based on the feistel network and dynamic DNA encoding, IEEE Photonics J. 10 (4) (2018) 1–14.
[17] W. Zhang, H. Yu, Z. Zhu, An image encryption scheme using self-adaptive selective permutation and inter-intra-block feedback diffusion, Signal Process. 151 (2018) 130–143.
[18] X. Wang, L. Liu, Y. Zhang, A novel chaotic block image encryption algorithm based on dynamic random growth technique, Opt. Lasers Eng. 66 (2015) 10–18.
[19] Y. Wang, K.W. Wong, X. Liao, et al., A new chaos-based fast image encryption algorithm, Appl. Soft Comput. 11 (1) (2011) 514–522.
[20] M. Alawida, A. Samsudin, J.S. Teh, et al., A new hybrid digital chaotic system with applications in image encryption, Signal Process. 160 (2019) 45–58.
[21] R. Lan, J. He, S. Wang, et al., Integrated chaotic systems for image encryption, Signal Process. 147 (2018) 133–145.
[22] X. Wang, H. Zhao, M. Wang, A new image encryption algorithm with nonlinear-diffusion based on Multiple coupled map lattices, Opt. Laser Technol. 115 (2019) 42–57.
[23] Q. Yin, C. Wang, A new chaotic image encryption scheme using breadth-first search and dynamic diffusion, Int. J. Bifurc. Chaos 28 (04) (2018) 1850047.
[24] X. Chai, Z. Gan, K. Yang, et al., An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations, Signal Process. Image Commun. 52 (2017) 6–19.
[25] H. Nematzadeh, R. Enayatifar, H. Motameni, et al., Medical image encryption using a hybrid model of modified genetic algorithm and coupled map lattices, Opt. Lasers Eng. 110 (2018) 24–32.
[26] S. Zhu, G. Wang, C. Zhu, A secure and fast image encryption scheme based on double chaotic s-boxes, Entropy 21 (8) (2019) 790.
[27] S. Zhu, C. Zhu, A new image compression-encryption scheme based on compressive sensing and cyclic shift, Multimed. Tools Appl. 78 (15) (2019) 20855–20875.
[28] Y. Li, C. Wang, H. Chen, A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation, Opt. Lasers Eng. 90 (2017) 238–246.
[29] H. Li, Y. Wang, Z. Zuo, Chaos-based image encryption algorithm with orbit perturbation and dynamic state variable selection mechanisms, Opt. Lasers Eng. 115 (2019) 197–207.
[30] E. Yavuz, A novel chaotic image encryption algorithm based on content-sensitive dynamic function switching scheme, Opt. Laser Technol. 114 (2019) 224–239.
[31] M. Wang, X. Wang, Y. Zhang, et al., A novel chaotic encryption scheme based on image segmentation and multiple diffusion models, Opt. Laser Technol. 108 (2018) 558–573.
[32] G. Cheng, C. Wang, H. Chen, A novel color image encryption algorithm based on hyperchaotic system and permutation-diffusion architecture, Int. J. Bifurc. Chaos 29 (09) (2019) 1950115.
[33] A.B. Cambel, Applied Chaos Theory: A Paradigm for Complexity, Elsevier, 1993.
[34] S. Vaidyanathan, C. Volos, Analysis and adaptive control of a novel 3-D conservative no-equilibrium chaotic system, Arch. Control Sci. 25 (3) (2015) 333–353.
[35] G. Qi, Modelings and mechanism analysis underlying both the 4D Euler equations and Hamiltonian conservative chaotic systems, Nonlinear Dyn. 95 (3) (2019) 2063–2077.
[36] M. Hénon, C. Heiles, The applicability of the third integral of motion: some numerical experiments, Astron. J. 69 (1964) 73.
[37] M. Lakshmanan, S. Rajaseekar, Nonlinear Dynamics: Integrability, Chaos and Patterns, Springer Science & Business Media, 2012.
[38] B. Eckhardt, G. Hose, E. Pollak, Quantum mechanics of a classically chaotic system: Observations on scars, periodic orbits, and vibrational adiabaticity, Phys. Rev. A 39 (8) (1989) 3776.
[39] J.C. Sprott, Some simple chaotic jerk functions, Am. J. Phys. 65 (6) (1997) 537–543.
[40] R. Thomas, Deterministic chaos seen in terms of feedback circuits: Analysis, synthesis," labyrinth chaos", Int. J. Bifurc. Chaos 9 (10) (1999) 1889–1905.
[41] S. Cang, A. Wu, Z. Wang, et al., Four-dimensional autonomous dynamical systems with conservative flows: two-case study, Nonlinear Dyn. 89 (4) (2017) 2495–2508.
[42] E. Dong, M. Yuan, S. Du, et al., A new class of Hamiltonian conservative chaotic systems with multistability and design of pseudo-random number generator, Appl. Math. Model. 73 (2019) 40–71.
[43] SIPI Image Database, University of Southern California Signal and Image Processing Institute (Accessed 29 May 2018). http://sipi.usc.edu/database/.
[44] E. Barker, A. RoginskyTransitions, Recommendation for transitioning the use of cryptographic algorithms and key lengths[J], NIST Special Publication 800 (2011) 131A.
[45] A. Yaghouti Niyat, M.H. Moattar, M. Niazi Torshiz, Color image encryption based on hybrid hyper-chaotic system and cellular automata, Opt. Lasers Eng. 90 (2017) 225–237.
[46] L. Xu, X. Gou, Z. Li, et al., A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion, Opt. Lasers Eng. 91 (2017) 41–52.
[47] C. Pak, L. Huang, A new color image encryption using combination of the 1D chaotic map, Signal Process. 138 (2017) 129–137.
[48] C. Zhu, G. Wang, K. Sun, Improved cryptanalysis and enhancements of an image encryption scheme using combined 1D chaotic maps, Entropy 20 (11) (2018) 843.
[49] C. Zhu, G. Wang, K. Sun, Cryptanalysis and improvement on an image encryption algorithm design using a novel chaos based S-box, Symmetry 10 (9) (2018) 399.
[50] L. Xu, Z. Li, J. Li, et al., A novel bit-level image encryption algorithm based on chaotic maps, Opt. Lasers Eng. 78 (2016) 17–25.
[51] Y. Zhou, W. Cao, C.L. Philip Chen, Image encryption using binary bitplane, Signal Process. 100 (2014) 197–207.