IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 29, 2019, accepted December 26, 2019, date of publication January 8, 2020, date of current version January 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963717

An Image Encryption Algorithm Based on Logistic-Fibonacci Cascade Chaos and 3D bit Scrambling

YUAN GUO, SHIWEI JING, YANYAN ZHOU, XIN XU, AND LIANSUO WEI
College of Computer and Control Engineering, Qiqihar University, Qiqihar 161006, China

Corresponding author: Shiwei Jing (2641235293@qq.com)

This work was supported in part by the National Science Foundation of China under Grant 61571150 and Grant 61872204, in part by the Chinese Education Department Overseas Returnees' Funds, in part by the Heilongjiang Provincial Natural Science Fund under Grant F2017029, and in part by the Heilongjiang Provincial Education Department through the Surface Scientific Research Project under Grant 135109236.

ABSTRACT Aiming at the problem that the existing bit scrambling encryption algorithm is not sensitive to the bit scrambling between 8 bits of one pixel, and the anti-noise and anti-selective plaintext attack ability is weak, a 3D cyclic shift bit scrambling image encryption algorithm is proposed in this paper. Discarding the scrambling mode of 8 bits of one pixel and the bits of all pixels, the pixel values are converted into binary arrays and then converted into 3D matrices in this paper. And the higher bit-planes and the lower bit-planes, which contain the plaintext information of different weights, are scrambled by cyclic shifting respectively. Therefore the sensitivity of the bit scrambling, the anti-noise attack ability of the algorithm and the randomness of intermediate ciphertext are improved. Moreover the randomness and the anti-noise ability can be adjusted by changing the number of higher bit-planes according to different encryption requirements. A new type of Logistic-Fibonacci (L-F) cascade chaos is constructed to generate random sequences, which solves the problem of blank windows in the uneven distribution of Logistic chaos. The initial value and the control parameters are increased, the sequence randomness is improved, and the fastness of low-dimensional chaos is preserved. By strongly correlating the key with SHA-256 of the plaintext, the key stream can change adaptively with the plaintext, which greatly improves the sensitivity of the plaintext and the ability of resisting the selective plaintext and ciphertext attack. The experiments show that the algorithm can encrypt all kinds of images with high efficiency, and can resist common attacks. It is a secure and reliable image encryption algorithm.

INDEX TERMS Image encryption, L-F cascade chaos, 3D bit scrambling, SHA-256.

I. INTRODUCTION

Images have the characteristics of intuition and vividness, which is widely used in the modern digital age. However, it is vulnerable to various attacks by hackers. For example, literature [1]–[4] propose a series of image copy-move forgery detection schemes. Therefore, more and more scholars pay attention to image security. Traditional encryption methods, such as DES, AES and IDEA, are suitable for text encryption and inefficient in image encryption [5], [6]. Some new encryption methods have been proposed, such as the encryption scheme of pixel position scrambling and pixel value replacement proposed by Chen G *et al.* [7] in 2004, which satisfies Shannon's confusion and diffusion principle of cryptographic design, and has attracted a lot of attention, improvement in literature [8]–[10]. However, because the chaotic sequence is not related with the plaintext, the encryption method is broken by choose plaintext/ ciphertext attack [11]–[13]. In 2010, Ye [14] proposed the idea of bit scrambling to convert the image into a two-dimensional binary matrix of M×8N. On this basis, the scrambling operation is performed, so that the pixel positions and the pixel values are changed in one operation, and the encryption algorithm complexity is reduced to save the encryption time. However, the security is low, and it is attacked by known plaintext, selected plaintext, and ciphertext only [15]–[17].

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

Moreover, the algorithm uses the row-column transformation, and does not change the proportion of 0, 1, so that the ciphertext distribution is not uniform enough. In the same year, Zhang *et al.* [18] added an XOR operation between different positions of the same picture after bit scrambling. This changes the 0 and 1 ratios to make the ciphertext distribution more uniform, but the encryption process becomes cumbersome and increases the encryption time. At the same time, the above two bit scrambling encryption methods are not good for some dark or white images, especially for pure white or pure black.

Recently, a large number of scholars have improved the above two types of encryption methods [19]–[25]. For example, Xiao-heng *et al.* [19] in 2014 and Guo-bo *et al.* [20] in 2016 added bit scrambling after pixel position scrambling on the basis of Literature [7], which not only retained the advantages of the original algorithm, but also further changed the pixel value through bit scrambling, better concealed the statistical characteristics of plaintext. By associating chaotic sequence with plaintext, the intermediate key adaptively changes with the plaintext, which makes it more effective to resist the selective plaintext attack, but it is still broken by Shuqin Zhu's selective plaintext attack [26]. Sun [21] added bit scrambling and DNA coding in Literature [7] in 2018, which increased the complexity of the algorithm and better resisted all kinds of attacks, but the over-complex algorithm increased the encryption time. The scrambling between 8 bits of one pixel in the literature [19]–[21] causes insensitivity to bit scrambling because it does not change the ratio of 0, 1 bit of each pixel value. That is to say, in the process of decryption, the outline of the plaintext can be seen without decrypting the bit scrambling. At the same time the encryption process is relatively cumbersome. In 2018, Gan *et al.* [22] used 3D bit scrambling, while Lv *et al.* [23] rearranged all binary numbers according to certain rules and then scrambled them. In this way, replacing the above pixel position scrambling and the scrambling between 8 bits of one pixel saves encryption time and overcomes the problem that the bit scrambling is not sensitive. However, the above improved algorithm scrambles the higher and lower bits together, which causes the higher bits to be switched to the lower, the ability to resist noise attack is reduced. At the same time, the method of pure row-column scrambling can't change the 0, 1 ratio of each row and column, the distribution of scrambled ciphertext is not uniform enough, and the ability of anti-statistical analysis is weak. Especially in Literature [23], in the process of plane scrambling, it is impossible to change the proportion of 0 and 1 per plane when the whole plane is scrambled together.

Chaotic system has good pseudo-randomness, ergodicity, high sensitivity to initial values and system parameters. Using chaotic system to generate intermediate key stream can reduce key volume, facilitate key transmission and distribution, and improve key sensitivity [27], [28]. Therefore, chaotic-based image encryption scheme is considered to be an image encryption scheme with great potential [29], [30]. A variety of chaos from one-dimensional logistic to five-dimensional hyperchaotic are also used in the above encryption algorithms. The one-dimensional chaotic logistic simple generation time is short, but there is a problem that the distribution is not uniform and the randomness is not high [31]. The high-dimensional chaotic dynamics are stronger, but the time-consuming is longer.

This paper constructs a new type of L-F cascade chaos, which overcomes the problem of uneven distribution while ensuring the rapidity of Logistic. The chaotic interval, initial value and parameters are enlarged to increase the key space and enhance the system's ability to resist brute force attacks. The randomness of the scrambled ciphertext is improved and the insensitivity of the bit scrambling is overcome by using the bit scrambling method of 3D cyclic shift. Separating the higher bit-planes containing a large amount of plaintext information from the lower bit-planes of a small amount of information improves the system's ability to resist noise attacks. The advantages and disadvantages of pixel scrambling and bit scrambling are compared through experiments. According to different encryption needs, the appropriate higher bit-planes number is selected, and the ciphertext randomness after scrambling and the anti-noise ability of the algorithm are coordinated. The encryption system uses SHA-256 of the plaintext to strongly correlate chaotic keys, which effectively improves the plaintext sensitivity and the ability to resist the selective plaintext attack.

II. ENCRYPTION SCHEME

The encryption algorithm in this paper includes three parts: L-F cascade chaos; key generation; 3D bit scrambling. The principles of these three parts are introduced below.

A. L-F CASCADE CHAOS

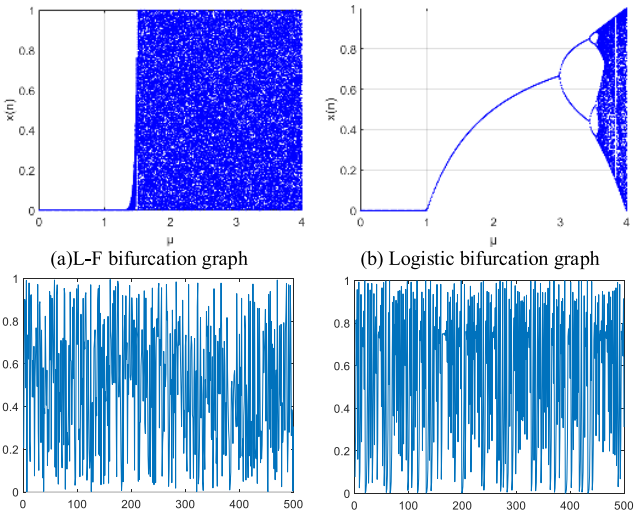
Logistic chaotic mapping is simple to express, easy to implement, and has good randomness. It is widely used in various fields of chaotic secure communication. The mapping is defined as [31]

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

The chaotic map is in a chaotic state when the parameter $\mu \in (3.57,4]$, and its value is distributed in the range of (0,1). In order to overcome the problem of blank windows in the uneven distribution of Logistic chaos, the Logistic and third-order Fibonacci cascade is used to make the full map and distribute more uniformly when the parameter μ is in the chaotic interval. The third-order Fibonacci function is as follows [32].

$$F_n = (AF_{n-1} + BF_{n-2} + CF_{n-3}) \bmod M \tag{2}$$

In (2) A, B, C and M represent constants, mod is modular operation. The L-F process takes the sequence x_n generated by Logistic as A, B and C in the third-order Fibonacci, and replaces each F with a set of A, B and C values. In order to make the initial value very sensitive, three initial values



(a)L-F bifurcation graph (b) Logistic bifurcation graph

(c) L-F Sequence distribution graph (d) Logistic sequence distribution graph

FIGURE 1. Bifurcation and sequence distribution diagrams of logistic and L-F.

TABLE 1. Logistic and L-F NIST SP800-22 randomness test.

Test name	L-F	Logistic
Frequency test	0.509663	0.057146
Block frequency test	0.545467	0.311542
Cumulative sums forward test	0.504531	0.000000
Cumulative sums reverse test	0.488341	0.000000
Runs test	0.498014	0.105618
Long runs test	0.474161	0.428497
Rank	0.452826	0.481624
FFT test	0.521041	0.45670
Non-overlapping templates test	0.501596	0.229238
Over-lapping templates test	0.461994	0.122079
Random excursions variant test	0.307696	0.000000
Approximate entropy test	0.500938	0.000000
Linear complexity test	0.509662	0.520064
Serial test	0.455003	0.000000
Universal test	0.534168	0.000000
Random excursions test	0.282841	0.000000

take the same value, namely $F_1 = F_2 = F_3 = \text{int } F$. The Equation is expressed as

$$F_n = (x_{n-1}F_{n-1} + x_{n-2}F_{n-2} + x_{n-3}F_{n-3}) \bmod M \quad (3)$$

In order to make the generated data consistent with the data range generated by Logistic, it is convenient to use in encryption, and then the generated sequence is mod 1, that is

$$B_n = F_n \bmod 1 \quad (4)$$

The bifurcation diagram of Logistic and L-F, the sequence distribution diagram and the random test of iteration 10^8 sequences, when $\mu = 3.999$, $x_0 = 0.5$, int $F = 0.9$ and $M = 191$, are shown in Figure 1 and Table 1.

Figure 1 shows that the L-F parameter range is wider than Logistic, and there is no blank window in the chaotic interval, the distribution is more uniform, and the number of initial values and control parameters is increased. The NIST test is a pass test as long as the probability is greater than 0.01. The greater the probability, the higher the randomness. From Table 1, we can see that all 16 tests of L-F have passed and are better than Logistic randomness. From the above analysis, it can be concluded that L-F is more secure for image encryption and can better withstand statistical analysis and brute force attack.

Chaos with different dimensions is generated on Intel (R) core (TM) i5-8500 CPU @ 3.00Hz, 8GB memory and win10 64 bit operating system computers. The average time of generating 10^6 sequence values with MATLAB 2016a is shown in Table 2.As can be seen from Table 2, L-F also retains the short time characteristic of logistic generating sequence.

B. KEY GENERATION

In this paper, there are two sets of keys $key1(x_{0,1}, \mu_1,$ int $F_{0,1}, k)$ and $key2(x_{0,2}, \mu_2,$ int $F_{0,2})$, which are the initial values and parameters of 3D bit scrambling and diffusion chaos respectively. For another parameter M in L-F chaotic system, the full text is set to 191. The k is the number of higher bit-planes scrambled in 3D, which is not associated with plaintext for convenience of direct control. Chaotic keys are connected with plaintext SHA-256, which makes the key stream adaptively change with plaintext. With the change of key stream, there is a strong avalanche effect between plaintext and ciphertext, which improves the ability of resisting selective plaintext and ciphertext attack and realizes "one plaintext, one key" and "one time, one key ". The 256-bit hash value of plaintext is divided into a group of eight bits, which can be expressed as $H = h_1, h_2, \cdots, h_{32}$, of which h_i is $h_i = [h_{i,0}, h_{i,1}, \cdots, h_{i,7}]$. The x and the int$F$ are generated according to the following.

$$\begin{aligned} x_{0,1} &= \text{mod}((x'_{0,1} + \text{mod}((h_1 \oplus h_2 \oplus h_3 \oplus h_4 \oplus h_5 \\ &\quad \times \oplus h_6 \oplus h_7 \oplus h_8), 256)/256), 1) \quad (5) \\ \text{int } F_{0,1} &= \text{mod}((\text{int } F'_{0,1} + \text{mod}((h_9 \oplus h_{10} \oplus h_{11} \oplus h_{12} \\ &\quad \times \oplus h_{13} \oplus h_{14} \oplus h_{15} \oplus h_{16}), 256)/256), 1) \quad (6) \\ x_{0,2} &= \text{mod}((x'_{0,2} + \text{mod}((h_{17} \oplus h_{18} \oplus h_{19} \oplus h_{20} \oplus h_{21} \\ &\quad \times \oplus h_{22} \oplus h_{24} \oplus h_{25}), 256)/256), 1) \quad (7) \\ \text{int } F_{0,2} &= \text{mod}((\text{int } F'_{0,2} + \text{mod}((h_{25} \oplus h_{26} \oplus h_{27} \oplus h_{28} \\ &\quad \times \oplus h_{29} \oplus h_{30} \oplus h_{31} \oplus h_{32}), 256)/256), 1) \quad (8) \end{aligned}$$

Among them, $\oplus$ is XOR operation, $x'_{0,1}, x'_{0,2} \in [0, 1]$, int $F'_{0,1}$, int $F'_{0,2} \in [0, 1]$. The control parameters μ_1 and μ_2 are generated by $x_{0,1}$, int $F_{0,1}$, $x_{0,2}$ and int $F_{0,2}$.

$$\mu_1 = \text{mod}(\mu'_1/4 + x_{0,1} + \text{int } F_{0,1}, 1) \times (4 - 1.5) + 1.5 \quad (9)$$
$$\mu_2 = \text{mod}(\mu'_2/4 + x_{0,2} + \text{int } F_{0,2}, 1) \times (4 - 1.5) + 1.5 \quad (10)$$

In (9) (10) the $\mu'_1, \mu'_2 \in [1.5, 4]$. $x'_{0,1}, x'_{0,2}$, int $F'_{0,1}$, int $F'_{0,2}$, μ'_1, μ'_2 and k can be set according to the need to control the encryption system conveniently.

TABLE 2. Time comparison table (units: s).

Logistic	L-F	Ref. [28] Coupled Tent Chaos	Ref. [22] Chen chaos	Ref. [18] Four-dimensional hyperchaos	Ref. [21] Five-dimensional hyperchaos
0.067250	0.100897	0.719836	2.624962	2.862689	1.461286

C. 3D BIT SCRAMBLING

The process of 3D bit scrambling is to convert plaintext image into binary sequence, and convert it into 3D matrix according to the number of artificially set higher bit-planes, and then scramble higher bit-planes and lower bit-planes separately.

1) CONSTRUCTION OF 3D BIT MATRIX

Before bit scrambling, image pixels need to be converted into binary digits and reconstructed. Here introduces the scrambling method of color image. Gray and binary images are not reconstructed. The color map has three channels of RGB. First, we separate the three channels and convert them into three $m \times n \times 8$ binary matrices. Put three-channel higher bit-planes together and lower bit- planes behind. Taking the number of higher bit-planes is 4 as an example, the first four bit-planes of B, G and R are placed on planes 1-4, 5-8 and 9-12 respectively, and then the last four bit-planes are placed on planes 13-16, 17-20 and 21-24 respectively to form a 3D matrix of $m \times n \times 24$. The conversion process of 3×3 color image is shown in Figure 2.

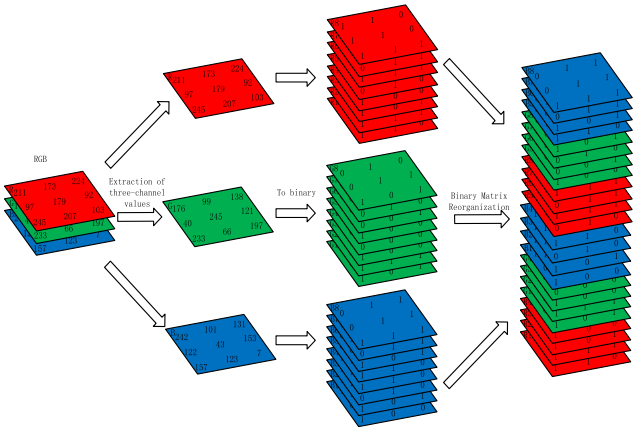


FIGURE 2. The process of image transformation into binary and reorganization.

2) BIT SCRAMBLING

In this paper, we adopt the scrambling method of right circular shift, such as [1, 2, 3, 4, 5, 6, 7, 8, 9] shifting 4 positions to [6, 7, 8, 9, 1, 2, 3, 4, 5]. The scrambling in this paper is mainly divided into two steps: Row and column cyclic shift scrambling according to the given random sequence on each plane; Row and column cyclic shift operation between the first $3 \times k$ planes and the last $3 \times (8-k)$ planes (for gray image and binary image, the first k planes and the last 8-k planes) respectively. In the case of $k = 4$, the two-step graphical process is given as shown in Figure 3 and Figure 4.

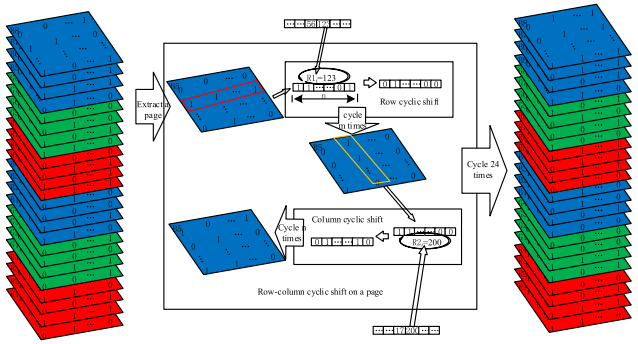


FIGURE 3. Row and column scrambling on each plane.

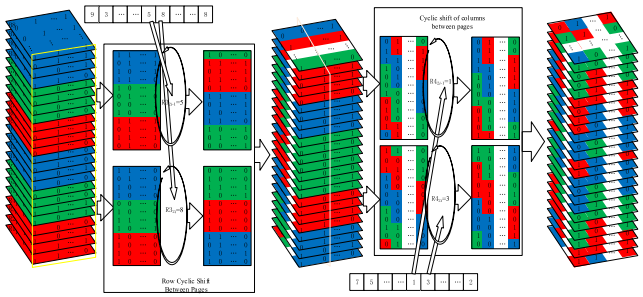


FIGURE 4. Row and column scrambling between planes.

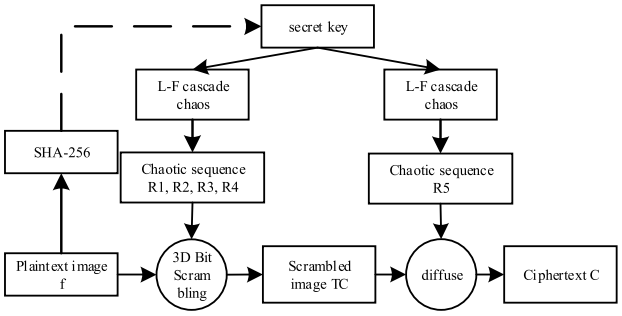


FIGURE 5. Encryption process.

In Figures 3 and Figure 4 the random sequences $\{1 \le R1 \le n \,|R1 \in N\}$, $\{1 \le R2 \le m \,|R2 \in N\}$, $\{1 \le R3 \le 3 \times k \,|R3 \in N\}$, $\{1 \le R4 \le 3 \times (8-k) \,|R4 \in N\}$ (in gray image encryption $\{1 \le R3 \le n \,|R3 \in N\}$, $\{1 \le R4 \le 8-k \,|R4 \in N\}$), where N is an integer, m, n is the number of rows and columns of the plaintext image, and k is the set number of higher bit-planes. After completing the scrambling operation mentioned above, a decimal noise-like intermediate ciphertext is obtained by 3D constructing reverse operation according to section II .C.1), which is prepared for the next diffusion operation.

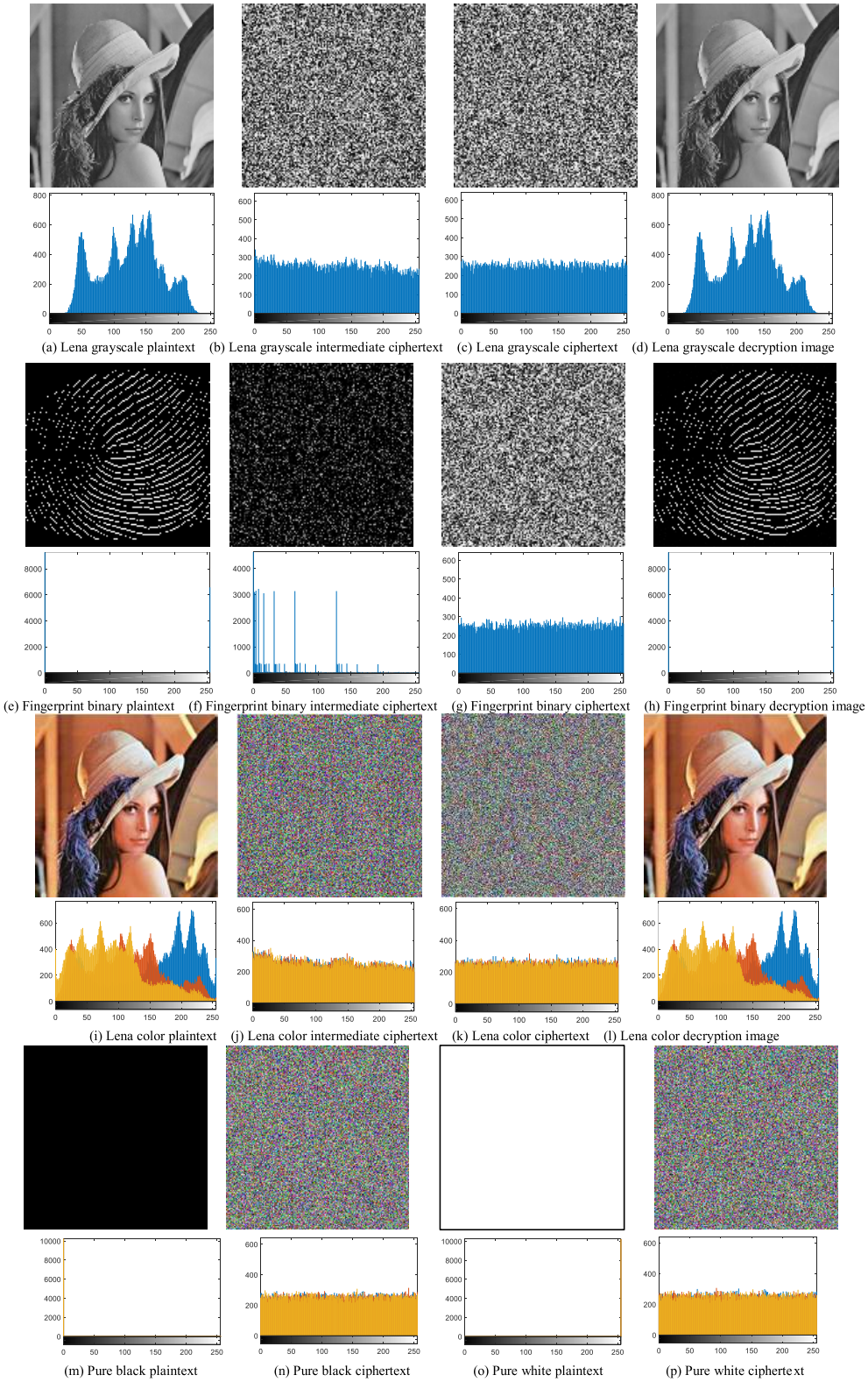


FIGURE 6. Encryption and decryption results.

III. IMAGE ENCRYPTION AND DECRYPTION PROCESSES

Image encryption is mainly divided into two steps: 3D bit scrambling encryption; diffusion operation. The encryption process is shown in Figure 5.

A. 3D BIT SCRAMBLING

In this paper, the main steps of 3D bit scrambling are shown in Section II C. First, an image f of $m \times n$ is read and converted into binary sequence, then a simple reorganization between

binary sequences is made into a 3D matrix. On this basis, bit scrambling is carried out and then converted into decimal intermediate ciphertext. Four random sequences $R1$, $R2$, $R3$, and $R4$ are generated by chaotic sequence L-F cascade chaos. In order to eliminate the transient effect, it is iterated $N_{0,1}$ times. $N_{0,1}$ is related to the initial value and parameter, the Equation is as shown in (11).

$$N_{0,1} = 200 + \left\lfloor \mathrm{mod}((x_{0,1} + \mathrm{int}\ F_{0,1} + \mu_1) \times 10^{12}, 200) \right\rfloor \quad (11)$$

$\lfloor\ \rfloor$ is upgraded. After iterating $N_{0,1}$ times, the chaotic sequence T can be obtained by continuing to iterate $26 \times (m + n)$ times (gray image only needs $10 \times (m + n)$ times, changing 24 of (12-13) to 8, changing $3 \times k$ of (14) to k, and changing $3 \times (8 - k)$ of (15) to $8 - k$). Equations (12-15) converts it into four random sequences $R1$, $R2$, $R3$ and $R4$ that are available for scrambling. The specific bit scrambling encryption process of color image is shown in Algorithm 1.

$$R1_i = \left\lfloor \mathrm{mod}(T_i \times 10^{12}, n) \right\rfloor + 1, \quad i = 1, 2, \cdots, 24 \times m \quad (12)$$

$$R2_i = \left\lfloor \mathrm{mod}(T_{24 \times m + i} \times 10^{12}, m) \right\rfloor + 1, \quad i = 1, 2, \cdots, 24 \times n \quad (13)$$

$$R3_i = \left\lfloor \mathrm{mod}(T_{24 \times (m+n) + i} \times 10^{12}, 3 \times k) \right\rfloor + 1 \quad i = 1, 2, \cdots, 2 \times m \quad (14)$$

$$R4_i = \left\lfloor \mathrm{mod}(T_{26 \times m + 24 \times n + i} \times 10^{12}, 3 \times (8 - k)) \right\rfloor + 1 \quad i = 1, 2, \cdots, 2 \times n \quad (15)$$

B. DIFFUSION OPERATION

3D bit scrambling changed the position of 0 and 1 bit, but did not change the overall proportion. The main purpose of diffusion operation is to change its proportion and better cover up the statistical characteristics of the image. At the same time, it also plays a diffusion role and increases the avalanche effect between plaintext and ciphertext.

The random sequence $R5$ used in the diffusion process is generated by L-F cascade chaos. In order to eliminate the transient effect, the sequence is obtained by iterating $N_{0,2}$ times, then $3 \times$ m $\times$ n times (gray image only needs $m + n$ times) and converted from (17) to integer sequence $R5$ between 0 and 255.

$$N_{0,2} = 200 + \left\lfloor \mathrm{mod}((x_{0,2} + \mathrm{int}\ F_{0,2} + \mu_2) \times 10^{12}, 200) \right\rfloor \quad (16)$$

$$R5_i = \left\lfloor \mathrm{mod}(T'_i \times 10^{12}, 256) \right\rfloor \quad i = 1, 2, \cdots\cdots \quad (17)$$

The random sequence $R5$ is transformed into a 3D matrix, which is the same as the intermediate ciphertext, in the order of row, column and plane, and then diffused by (18-20). If it is grayscale image encryption, it only needs to use the grayscale image as the R channel of the color map, and bring it into the formula (18) to complete the diffusion encryption operation.

$$CR(i,j) = \begin{cases} \mathrm{mod}(TCR(i,j) + R5R(i,j), 256) \oplus CR0, \\ i = 1, j = 1 \\ \mathrm{mod}(TCR(i,j) + R5R(i,j), 256) \oplus CR(i-1, n), \\ 1 < i \le m, j = 1 \\ \mathrm{mod}(TCR(i,j) + R5R(i,j), 256) \oplus CR(i, j-1), \\ 1 < j \le n \end{cases} \quad (18)$$

Algorithm 1 3D bit Scrambling

Input: Color images f of size m × n and key1

1: Converting f into a 3D matrix P of m × n × 24 according to the 3D conversion mode of section II .C.1.

2: The number of pre-iterations is calculated by key1 and (11) and the chaotic sequence T is obtained by introducing L-F cascade chaotic system.

3: The chaotic sequences R1, R2, R3 and R4 for scrambling are obtained by using (12) (13) (14) (15).

4: Scrambling by following pseudocode

```
for k1=1:24
    for i1=1:m
        tem(i1,:, k1)=cirshift(P(i1,:, k),
        R1((k1-1)*m+i1));
    end
    for j1=1:n
        tem1(:, j1, k1)=cirshift(tem(:, j1, k),
        R2((k1−1)*n+j1));
    end
end
for i2=1:m
        tem2(i2,:,:)=cat(3, cirshift(tem1(i2,:,1:3*k),
                            R3(2*i2-1)),
                            cirshift(tem1(i2,:,3*k+1:24),
                            R3(2*i2)));
end
for j2=1:n
        tem3(:, j2,:)=cat(3, cirshift(tem2(:, j2,1:3*k),
                           R4(2*j2-1)),
                           cirshift(tem2(:, j2,3*k+1:24),
                           R4(2*j2)));
end
```

5: The decimal intermediate ciphertext TC is obtained by reverse operation of tem3 in section II .C.1) 3D conversion mode.

Output: Intermediate ciphertext TC.

In the pseudocode above, a = cirshift (b, c) shifts array b to the right in step c and assigns the result to a. a = cat (3, b, c) is a new matrix composed of three-dimensional matrices b and c by overlapping planes, and assigned to a.

$$CG(i,j)=\begin{cases}\mathrm{mod}(TCG(i,j)+R5G(i,j),256)\oplus CG0,\\ i=1,j=1\\ \mathrm{mod}(TCG(i,j)+R5G(i,j),256)\oplus CG(i-1,n),\\ 1<i\le m,j=1\\ \mathrm{mod}(TCG(i,j)+R5G(i,j),256)\oplus CG(i,j-1),\\ 1<j\le n\end{cases} \tag{19}$$

$$CB(i,j)=\begin{cases}\mathrm{mod}(TCB(i,j)+R5B(i,j),256)\oplus CB0,\\ i=1,j=1\\ \mathrm{mod}(TCB(i,j)+R5B(i,j),256)\oplus CB(i-1,n),\\ 1<i\le m,j=1\\ \mathrm{mod}(TCB(i,j)+R5B(i,j),256)\oplus CB(i,j-1),\\ 1<j\le n\end{cases} \tag{20}$$

CR, *CG* and *CB* are R, G and B channels of ciphertext respectively. Similarly, *TCR*, *TCG*, *TCB* and *R5R*, *R5G*, *R5B* are three channels of intermediate ciphertext and random sequence converted to 3D matrix. *CR0*, *CG0* and *CB0* are the initial values of diffusion. *CR0* uses *key*2 to generate in (21) while *CG0* and *CB0* are $CR(m,n)$ and $CG(m,n)$, respectively. This makes the latter ciphertext affected by the former ciphertext. The former channel of the three channels also affects the latter channel ciphertext generation. Compared with the way of setting an initial value for each of the three channels in reference [23], the influence between the three channels is more obvious. The specific steps of diffusion operation of color image are shown in Algorithm 2.

$$CR0=\left\lfloor \mathrm{mod}((x_{0,2}+\mathrm{int}\ F_{0,2}+\mu_2)\times 10^{12},256)\right\rfloor \tag{21}$$

C. DECRYPTION PROCESS

The decryption process is the inverse process of the encryption process. The intermediate ciphertext *TC* is obtained by using (22-24) (Grayscale images only need to be brought in (22)) for diffusion decryption.

$$TCR(i,j)=\begin{cases}\mathrm{mod}(CR(i,j)\oplus CR0+256-R5R(i,j),256),\\ i=1,j=1\\ \mathrm{mod}(CR(i,j)\oplus CR(i-1,n)+256-R5R(i,j),256),\\ 1<i\le m,j=1\\ \mathrm{mod}(CR(i,j)\oplus CR(i,j-1)+256-R5R(i,j),256),\\ 1<j\le n\end{cases} \tag{22}$$

$$TCG(i,j)=\begin{cases}\mathrm{mod}(CG(i,j)\oplus CG0+256-R5G(i,j),256),\\ i=1,j=1\\ \mathrm{mod}(CG(i,j)\oplus CG(i-1,n)+256-R5G(i,j),256),\\ 1<i\le m,j=1\\ \mathrm{mod}(CG(i,j)\oplus CG(i,j-1)+256-R5G(i,j),256),\\ 1<j\le n\end{cases} \tag{23}$$

Algorithm 2 Diffusion Operation

Input: Intermediate ciphertext TC, key2

1: The number of pre-iterations is calculated by using key2 and (16), which are brought into L-F cascade chaotic system to obtain chaotic sequence T'.

2: The chaotic sequence R5 used for scrambling is obtained by using (17) and converted into a 3D matrix like the intermediate ciphertext.

3: The initial value CR0 of diffusion is obtained by using key2 and (21).

4: Diffusion operations are performed according to the following pseudocode

```
for k=1:3
    switch k
        case 1
            cc=CR0;
        case 2
            cc=C(m,n,1);
        otherwise
            cc=C(m,n,2);
    end
    for i1=1:m
        for j1=1:n
            if (i1==1 && j1==1)
          C(i1,j1,k)=bitxor(mod(TC(i1,j1,k)
          +R5(i1,j1,k),256),cc);
            elseif (i1~=1 && j1==1)
        C(i1,j1,k)=bitxor(mod(TC (i1,j1,k)
        +R5(i1,j1,k),256),C(i1-1,n,k));
            else
        C(i1,j1,k)=bitxor(mod(TC (i1,j1,k)
        +R5(i1,j1,k),256),C(i1,j1-1,k));
            end
        end
      end
    end
```

Output: Ciphertext C.

$$TCB(i,j)=\begin{cases}\mathrm{mod}(CB(i,j)\oplus CB0+256-R5B(i,j),256),\\ i=1,j=1\\ \mathrm{mod}(CB(i,j)\oplus CB(i-1,n)+256-R5B(i,j),256),\\ 1<i\le m,j=1\\ \mathrm{mod}(CB(i,j)\oplus CB(i,j-1)+256-R5B(i,j),256),\\ 1<j\le n\end{cases} \tag{24}$$

The intermediate ciphertext is converted into 3D binary matrix according to section II .C.1). Firstly, *R*4 and *R*3 are used to transform the left cyclic shift of column-rows between planes. Then, *R*2 and *R*1 are used to transform the left cyclic shift of column-row for each plane. Finally, transform it according to the inverse of Section II .C.1) convert to decimal decrypted image f.

IV. EXPERIMENTAL ANALYSIS

In order to verify the validity and feasibility of the proposed algorithm, MATLAB R2016a is used as the simulation platform, and grayscale, binary image, color image, and pure black white image are selected as plaintext. Set $x'_{0,1}$, $x'_{0,2}$, int $F'_{0,1}$, int $F'_{0,2}$, μ'_1, μ'_2 and k to 0.9, 0.95, 0.9, 0.95, 1.6, 1.65 and 3 respectively, and then generate different keys according to SHA-256 of different encrypted images. The results of encryption and decryption and the corresponding histogram are shown in Figure 6.

Figure 6 shows that the intermediate ciphertext and ciphertext of gray image, binary image and color image are noise-like images. The histogram of the intermediate ciphertext has been changed compared with the plaintext, and the ciphertext histogram has been fairly smooth, which well conceals the plaintext information. For pure black and white images, because bit scrambling can't change the proportion of 0 and 1 bits, so the intermediate ciphertext is the same as plaintext, but the ciphertext histogram is smooth, the distribution of pixel values is uniform, so this paper can encrypt all kinds of images very well. The decrypted image under the correct key is exactly the same as the plaintext, which shows that the decryption effect is good.

A. COMPARISON AND ANALYSIS OF ANTI-NOISE ABILITY AND DISTRIBUTION OF 3D BIT SCRAMBLED CIPHERTEXT

In this paper, when 3D bit scrambling, the planes with more plaintext information are scrambled separately from the planes with less plaintext information. When the number of higher bit-planes of the three channels is increased from 0 to 8, the information entropy (H) of the ciphertext after scrambling and the correlation coefficient (CC) between the plaintext and decrypted image of the ciphertext with the Gaussian noise of intensity 0.1, are as shown in the Figure 7. The higher bit-plane number is 0 as a special case, and the right circular shift is used to perform pixel value scrambling, so as to compare the difference between bit scrambling and pixel scrambling. Equation (25-27) respectively obtains image information entropy, adds Gaussian noise to the image, and CC of the two images.

$$H = \sum_{i=0}^{255} p(i) \log \frac{1}{p(i)} \tag{25}$$

$$rm = \text{uint8}(rm \times (1 + k1G)) \tag{26}$$

$$CC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[f(x,y)-\overline{f}][F(x,y)-\overline{F}]}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}[f(x,y)-\overline{f}]^2}\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}[F(x,y)-\overline{F}]^2}} \tag{27}$$

Among them, $p(i)$ is the probability when the pixel value is i. $k1$ is the noise intensity. G is Gaussian noise of the mean value is 0 and variance is 1. uint8(x) is rounded-off when $x \in [0, 255]$, 0 when $x < 0$, and 255 when $x > 255$. $f(x, y)$ as plaintext, $F(x, y)$ as decrypted image, $\overline{f}$ and $\overline{F}$ as the mean of corresponding image pixels.

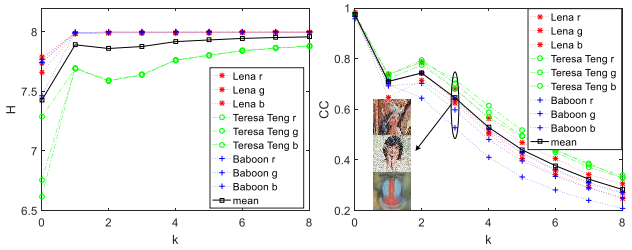


FIGURE 7. The ciphertext information entropy and the CC between decryption image and plaintext.

Figure 7 is obtained with scrambling only and no diffusion process. The scrambling of $k = 0$ visible pixels does not change the pixel value, so the information entropy is relatively low, but it has the strongest anti-noise ability. Bit scrambling anti-noise ability is reduced, but the plaintext information can be clearly seen when the noise intensity is not too large. At the same time, it changes the pixel value while changing the position, and the ciphertext information entropy is greatly improved. Therefore, bit scrambling is still very practical. It can be seen from Figure 7 that in most cases, the information entropy will increase with the increase of k, except for some images with extremely uneven ratios on some planes of 0 and 1 bits, such as Teresa Teng, but they are much higher than the pixel scrambling. The anti-noise ability generally decreases as k increases, but anomalies occur when $k = 1$, which is mainly due to the fact that the second bit-plane with a large amount of plaintext information participates in the lower bit-planes scrambling. This paper can adjust k according to different encryption requirements for information entropy and anti-noise ability. Since $k = 3$ has better information entropy and strong anti-noise attack ability, this paper selects $k = 3$ without special explanation.

B. SENSITIVITY ANALYSIS OF BIT SCRAMBLING

The scrambling of 8 bits of one pixel is not sensitive in the bit scrambling process, that is, the plaintext information can be seen without performing bit scrambling in the decryption process. In this paper, we use 3D bit scrambling to solve the problem. Since the comparative literature uses grayscale images, this section of the plaintext image uses grayscale images, and the decryption image of our encryption algorithm and reference [19]–[21] without bit scrambling in the decryption process are shown in Figure 8.

From Figure 8, it can be seen that the image decrypted without bit scrambling in reference [19]–[21] can still see the outline of plaintext image. The image with high contrast, such as Teresa Teng, has worse encryption effect and completely loses effect when binary image appears. In these three cases, the proposed encryption algorithm can't see the plaintext information at all. Therefore, the encryption scheme in this paper is more sensitive to the bit scrambling.

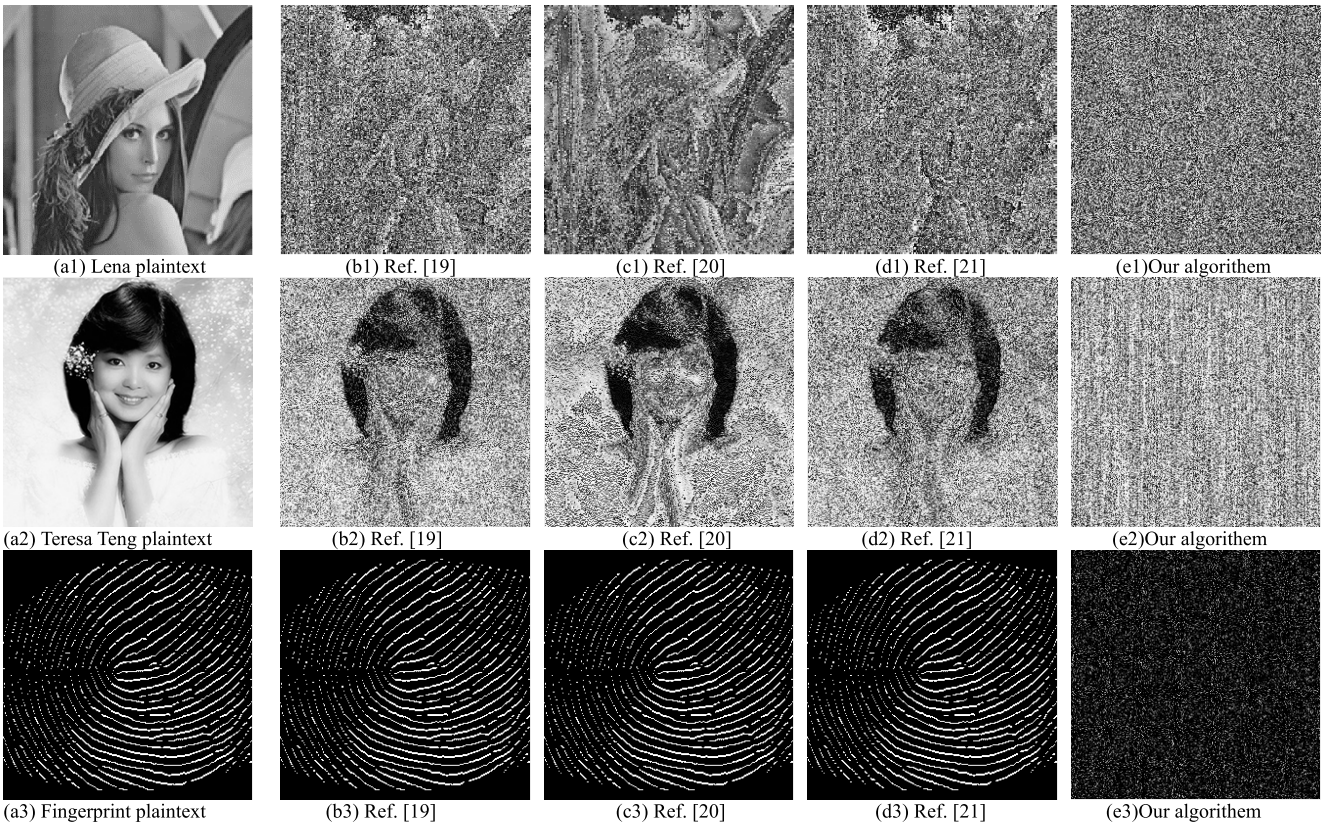


FIGURE 8. Sensitivity analysis of chaotic sequences used for bit scrambling.

FIGURE 9. Clipped ciphertext and its decrypted image.

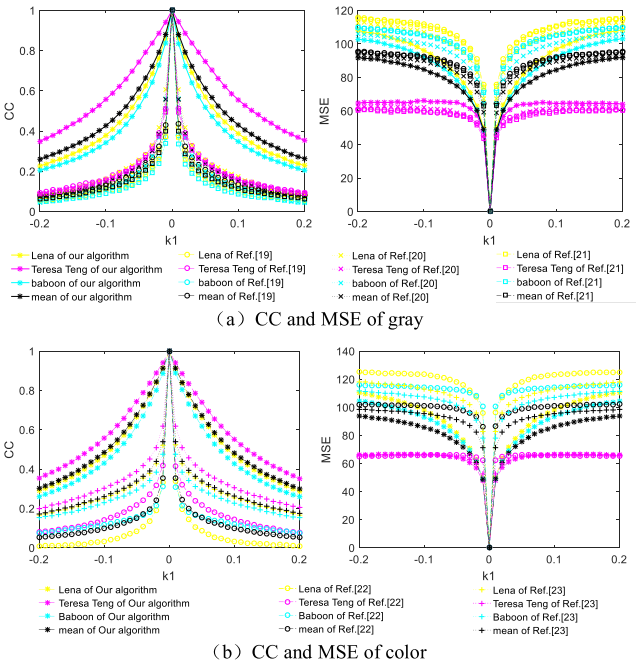


FIGURE 10. Comparison of CC and MSE.

C. ROBUSTNESS ANALYSIS

In the process of ciphertext transmission, ciphertext is vulnerable to information loss and pollution, so a good encryption algorithm must have the ability to resist crop and noise attacks. Next, the anti-cropping and anti-noise attacks of encryption algorithms are compared and analyzed.

1) CROPPING ATTACK

Figure 9 shows the areas of 1/16, 1/8, 1/4, and 1/2 cut in the ciphertext image when the color image Lena (256*256) is used as the plaintext. And give the corresponding decrypted image. As can be seen from Figure 9, when 50% of the data is lost, the decrypted image can still be recognized and contains most plaintext information, which shows that the encryption algorithm has strong robustness against cropping attacks.

To compare the anti-cropping ability, the *CC* between decrypted and plaintext images with the same size of

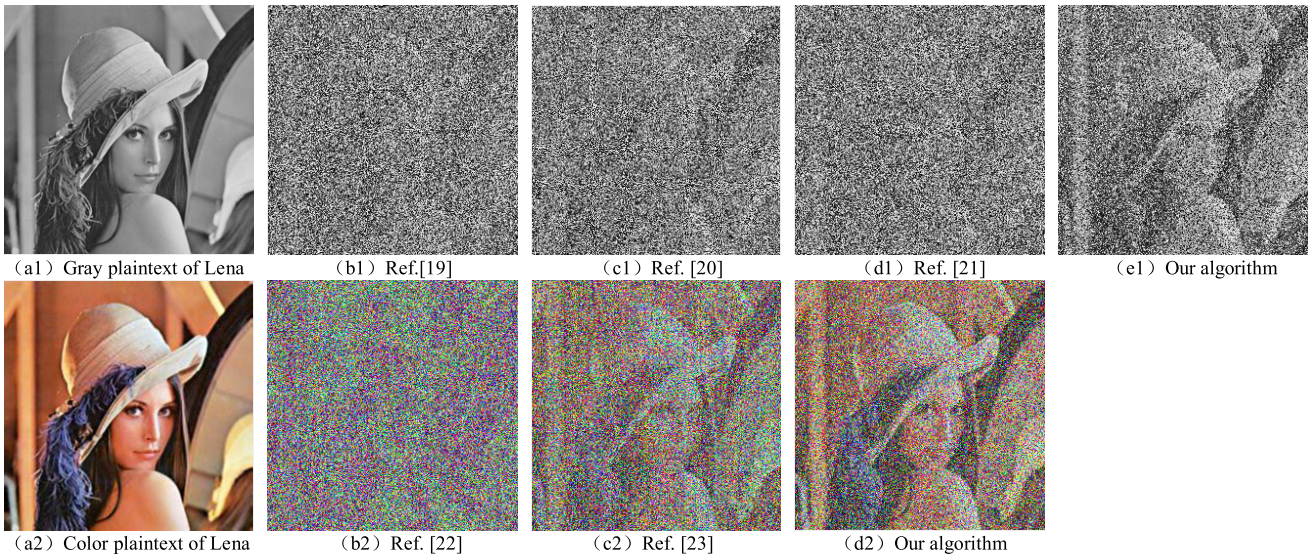


FIGURE 11. Decrypted image with noise added to ciphertext.

ciphertext cropping is calculated as shown in Table 3. In the comparison literature, there are color image encryption and grayscale image encryption. Therefore, the gray and color maps of Lena and Teresa Teng are used as plaintext and the ciphertext is cut 50% to obtain the decrypted image, and the *CC* is obtained with between decrypted and plaintext images.

In Table 3, the *CC* of gray scale image is the mean of two images and the *CC* of color image is the mean of three channels of two images. Table 3 shows that the *CC* between decrypted image and plaintext can reach 0.4 when 50% of the ciphertext is crop, which is better than reference [19]–[23]. It shows that the algorithm has a good ability to resist the cut attack.

2) NOISE ATTACK

In this paper, Equation (26) is used to add Gauss noise to ciphertext, and mean square error (*MSE*) and *CC* are used to evaluate the quality of decryption. The smaller the *MSE* and the closer the *CC* to 1, the smaller the difference between decrypted image and plaintext, the better the decryption effect and the better the robustness of the encryption system.

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} [F(x, y) - f(x, y)]^2 \qquad (28)$$

The *CC* and *MSE* between the decrypted image and plaintext are calculated after adding noise of different intensity as shown in Figure 10.

MSE and *CC* of the color image in Figure 10 are the average values of the RGB three channels. It can be seen from Figure 10 that with the change of the noise intensity $k1$, the *CC* of the proposed algorithm is closest to nearly 1 and the *MSE* is the smallest, so the algorithm has the strongest anti-noise attack capability. In order to visually see the difference, the image is decrypted when $k1 = 0.15$ and plaintext as Lena, as shown in Figure 11.

It can be seen from Figure 11 that the encryption algorithm has the clearest decryption image under the same intensity noise attack, and retains more detailed information. In the comparative literature, the higher bits and the lower bits are scrambled together, and the anti-noise attack capability is significantly reduced. In particular, the reference [22] scrambles the entire plane between 24 planes when the 3D Scrambling, and the ability to resist noise attacks is highly dependent on these 24 random numbers. At the same time, in [22], in order to improve the interaction between the three channels of RGB, the XOR between different channels is increased during the diffusion, so that the items related to ciphertext in the decryption process are one more than the proposed algorithm and the reference [23]. This reduces it anti-noise attack capability again. The proposed algorithm uses high and low binary planes to separate scrambling, so that high-order binary will not appear in the low-order planes. At the same time, the row and column are performed twice in the process of scrambling between planes, and the planes of each corresponding row and column are equivalent to the sum of the cyclic shifts twice, so that the number of cyclic shifts between planes corresponding to each row and column different, there will be no case where the same plane is changed to another planes. This scrambling process greatly improves the noise immunity of the proposed algorithm.

Through the analysis of the crop attack and the noise attack, it can be seen that the encryption algorithm greatly improves the anti-noise attack ability without reducing the anti-crop attack capability, so the proposed algorithm is more robust.

D. STATISTICAL CHARACTERISTIC ANALYSIS

This paper mainly analyses the statistical characteristics from two aspects. One is the distribution of image pixel values and the degree of value confusion, that is, histogram and entropy.

TABLE 3. CC between decrypted image and plaintext after 50% ciphertext cropping.

Our algorithem(gray)	Ref. [19]	Ref. [20]	Ref. [21]	Our algorithem(color)	Ref. [22]	Ref. [23]
0.4358	0.4335	0.4294	0.4294	0.4753	0.4621	0.4621

TABLE 4. The χ^2 value of histogram of gray scale images.

Algorithms		Lena (256*256)	Fingerprint (256*256)
Plaintext		40483.44	13682610
Ref.[19]	Intermediate ciphertext	4223.56	13682610
	Ciphertext	236.34	268.39
Ref.[20]	Intermediate ciphertext	19138.57	13682610
	Ciphertext	274.19	236.34
Ref.[21]	Intermediate ciphertext	6193.25	13682610
	Ciphertext	289.39	226.25
Our algorithm	Intermediate ciphertext	592.78	3242350.46
	Ciphertext	236.34	221.48

TABLE 5. The χ^2 value of histogram of color scale images.

Algorithms		Lena (256*256*3)	White (256*256*3)	Black (256*256*3)
Plaintext		23113.47	16711680	16711680
Ref.[22]	Intermediate ciphertext	24895.87	16711680	16711680
	Ciphertext	252.22	241.16	243.61
Ref.[23]	Intermediate ciphertext	2907.35	16711680	16711680
	Ciphertext	244.50	235.32	243.44
Our algorithm	Intermediate ciphertext	744.38	16711680	16711680
	Ciphertext	237.95	233.06	236.21

The other is the degree of correlation between adjacent pixels and three channels of color image.

1) HISTOGRAM ANALYSIS

Histogram has been given in Figure 6, next we mainly use χ^2 of the histograms to quantitatively reflect the uniformity of histogram. The χ^2 value of histograms of the corresponding plaintext, intermediate ciphertext and ciphertext in Figure 6 are given in Table 4 and Table 5. In order to compare the encryption effect of other algorithms, Table 4 and Table 5 also gives the χ^2 value of the histogram of the intermediate ciphertext and ciphertext of other algorithms. χ^2 of histograms of an image with 256 gray levels can be computed by [34]

$$\chi^2 = \sum_{i=0}^{255} \frac{(z_i - kk/256)^2}{kk/256} \tag{29}$$

where $z = \{z_0, z_1, \cdots, z_{255}\}$ is the vector of the histogram values, and z_i is the numbers of pixels which gray values are equal to i. $kk/256$ is the expected occurrence frequency of each gray level and kk is the number of all the pixels.

The χ^2 of the histograms of color image in Table 5 are the mean values of three channels. When the significant level are 0.05 and 0.1, the corresponding $\chi^2(0.05, 255)$ and $\chi^2(0.1, 255)$ are 293.25 and 284.34, respectively. The smaller the value of the χ^2, the more uniform the histogram distribution. If the χ^2 value is lower than 293.25, it passes the χ^2 test [34]. It can be seen from Table 4 and Table 5 that the χ^2 value of the intermediate ciphertext in the proposed algorithm is significantly lower than that in reference [19]–[23] when using Lena and Fingerprint images, so the histogram distribution of the intermediate ciphertext in the proposed algorithm is more uniform and the encryption effect is better. At the same time, the χ^2 value of ciphertext is not only less than 293.25 but also less than 284.34, so the proposed algorithm can pass the χ^2 test well.

2) ENTROPY ANALYSIS

Information Entropy is used to represent the overall randomness of a picture. The ideal value of information Entropy is 8 for data of uint8 type. If the information entropy of ciphertext image is less than 8, it may be attacked by a certain degree of predictability. The information entropy is shown in (25) and the results are shown in Table 6 and Table 7.

From Table 6 and Table 7, it can be seen that the randomness of ciphertext distribution of the encryption algorithm proposed in this paper is consistent with existing literatures and can reach 7.99, which is very close to the ideal value of 8, this shows that the randomness of ciphertext is very high. The intermediate ciphertext of this algorithm is also distributed evenly. In references only the Reference [23] is higher than our algorithm, and the gap is not very large, unlike the other comparative, the gap reaches one decimal place. The contrast reference in gray image only scrambles the 8-bit binary of a single pixel, and does not change the 0-1 ratio of a pixel. Obviously, the scrambling effect is not good together. Reference [22] in color image is 3D scrambling, but when it scrambles between planes, it scrambles the whole plane together. It can't change the 0-1 ratio of each plane, which makes the information entropy of intermediate ciphertext not high enough. Our algorithm uses 3D scrambling not only changes the ratio of 0 and 1 per-pixel, but also changes the ratio of 0 and 1 per-plane. At the same time, the algorithm can also adjust the number of higher bit-planes to further improve the information entropy. There is still a certain advantage in the randomness of the intermediate ciphertext.

The local Shannon entropy [33] is used to measure the local randomness of the image. It is divided into three steps to obtain: firstly, the regions $M_1, M_2, \ldots, M_k$ in which the k blocks are not repeated are taken in the test image, where in each region has T_B pixel values; then the information of each region is calculated by using (25). Finally, the local Shannon entropy is obtained by using (30).

$$\overline{H}_{k,T_B} = \sum_{i=1}^{k} \frac{H(M_i)}{k} \tag{30}$$

TABLE 6. Information entropy comparison of gray images.

Images	Plaintext	Our algorithm		Ref.[19]		Ref. [20]		Ref. [21]	
		Intermediate ciphertext	Ciphertext	Intermediate ciphertext	Ciphertext	Intermediate ciphertext	Ciphertext	Intermediate ciphertext	Ciphertext
Lena(256*256)	7.4442	7.9932	7.9975	7.9429	7.9975	7.7491	7.9970	7.9248	7.9967
Teresa Teng(256*256)	6.7901	7.5992	7.9971	7.7517	7.9976	7.2786	7.9967	7.6969	7.9971
Baboon(512*512)	7.3609	7.9982	7.9993	7.9592	7.9992	7.6905	7.9994	7.9473	7.9993
Fingerprint(256*256)	0.4701	3.7584	7.9977	0.4701	7.9971	0.4701	7.9975	0.4701	7.9976
Mean	4.4131	6.8372	7.9979	6.031	7.9979	5.7971	7.9977	6.0098	7.9977

TABLE 7. Information entropy comparison of color images.

Images		Plaintext	Our algorithm		Ref.[22]		Ref. [23]	
			Intermediate ciphertext	Ciphertext	Intermediate ciphertext	Ciphertext	Intermediate ciphertext	Ciphertext
Lena (256*256)	R	7.7394	7.9918	7.9971	7.6930	7.9974	7.9586	7.9974
	G	7.7910	7.9925	7.9973	7.9424	7.9971	7.9702	7.9976
	B	7.6600	7.9902	7.9968	7.1380	7.9969	7.9747	7.9976
Teresa teng (256*256)	R	6.6188	7.7605	7.9968	6.1174	7.9972	7.8063	7.9972
	G	6.7561	7.7600	7.9974	6.7715	7.9973	7.8735	7.9972
	B	7.2857	7.7659	7.9971	7.3323	7.9972	7.8753	7.9970
Baboon (512*512)	R	7.7432	7.9985	7.9993	7.2801	7.9994	7.9974	7.9993
	G	7.4662	7.9985	7.9994	7.9895	7.9993	7.9971	7.9993
	B	7.7683	7.9986	7.9992	7.7876	7.9993	7.9980	7.9993
White (256*256)	R	0	0	7.9973	0	7.9971	0	7.9971
	G	0	0	7.9972	0	7.9971	0	7.9969
	B	0	0	7.9972	0	7.9976	0	7.9974
Black (256*256)	R	0	0	7.9972	0	7.9971	0	7.9971
	G	0	0	7.9971	0	7.9972	0	7.9971
	B	0	0	7.9973	0	7.9969	0	7.9971
Mean		4.4552	4.7504	7.9976	4.4035	7.9976	4.7634	7.9976

The $H(M_i)$ is the information entropy of the M_i submodule. In this paper, 20 submodules are not repeated in the image to be tested, and each module has 1936 pixels. The results of calculating the local Shannon entropy are shown in Table 8.

It can be seen from Table 8 that the average local Shannon entropy of ciphertext is greater than 7.9 and the plaintext is less than 7, which means that the ciphertext image obtained by our algorithm has good local randomness and can resist entropy attack.

3) CORRELATION ANALYSIS

The adjacent pixels of the plaintext image have high correlation in the horizontal, vertical and diagonal directions, and the attacker can recover the image by analyzing the related information. Therefore, an effective encryption algorithm should remove these pixel correlations and generate low-correlation ciphertext. In order to visually see the correlation of adjacent pixels, 10000 horizontal adjacent pixel relationship diagrams of the gray Lena image and the color Lena image of the proposed encryption algorithm are given, as shown in Figure 12. The *CC* between the plaintext image and the ciphertext calculated by (27) is shown in Table 9.

Figure 12 shows that plaintext is mainly distributed on the diagonal line, indicating that its adjacent pixels are basically the same, while the distribution of intermediate ciphertext and ciphertext is very uniform, indicating that the algorithm destroys the correlation of adjacent pixels of plaintext very well. It can be seen from Table 9 that the effective

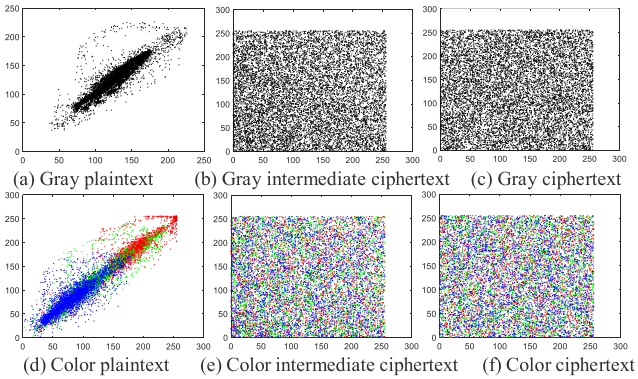


FIGURE 12. Lena image horizontal direction adjacent pixel correlation.

coefficient of the CC of the ciphertext of the proposed algorithm reaches the last three decimal places are very close to 0. Compared with the existing bit scrambling, there is a very low correlation even more than the Reference [20]–[22]. Well, the correlation of the ciphertext of the proposed algorithm has been very low.

The algorithm can also eliminate the high correlation among R, G and B components of color image. The R, G and B components of Lena color image plaintext, intermediate ciphertext and ciphertext are extracted from 10000 pixels in the same position to draw the correlation figure of pixels, as shown in Figure 13. At the same time, Table 10 lists the

TABLE 8. Local shannon entropy.

Image	Color (plaintext)			Color (ciphertext)			Gray (plaintext)	Gray (ciphertext)
	R	G	B	R	G	B		
Lena (256*256)	6.8272	6.8768	6.8150	7.9065	7.9048	7.908	6.5395	7.9075
Teresa teng (256*256)	5.4067	5.3938	5.9393	7.9054	7.9060	7.9096	5.4636	7.9037
Baboon (512*512)	6.9023	6.8613	6.9762	7.9072	7.9097	7.9089	6.7819	7.9088
fingerprint (256*256)	—	—	—	—	—	—	0.4594	7.9088
White (256*256)	0	0	0	7.9055	7.9052	7.9084	—	—
Black (256*256)	0	0	0	7.9060	7.9052	7.9067	—	—
Mean		3.8666			7.9069		4.8111	7.9072

TABLE 9. Pixel adjacent pixel correlation.

		Horizontal	Vertical	Diagonal	Mean |CC|
Lena (gray)		0.9357	0.9682	0.9085	0.9375
Our algorithm		-0.0032	-0.0024	0.0033	0.0030
Ref.[19]		0.0017	-0.0003	0.0001	0.0007
Ref. [20]		0.0015	-0.0021	0.0075	0.0037
Ref. [21]		0.0068	-0.0054	0.0010	0.0044
Lena (color)	R	0.9635	0.9824	0.9423	
	G	0.9551	0.9800	0.9331	0.9568
	B	0.9512	0.9761	0.9273	
Our algorithm	R	0.0055	0.0051	-0.0005	
	G	-0.0013	0.0023	-0.0055	0.0029
	B	0.0026	-0.0004	0.0033	
Ref. [22]	R	0.0066	0.0065	-0.0308	
	G	0.0291	0.0191	-0.0140	0.0130
	B	-0.0052	-0.0003	0.0056	
Ref. [23]	R	0.0001	0.0001	0.0003	
	G	-0.0016	-0.0003	0.0002	0.0006
	B	0.0009	0.0007	-0.0012	

TABLE 10. The same location correlation of three channels.

	R-G	R-B	G-B	Mean (|CC|)
Lena	0.8829	0.6944	0.9216	0.8330
Our algorithm	-0.0008	0.0010	-0.0045	0.0021
Ref.[22]	-0.0111	-0.0003	0.0082	0.0065
Ref. [23]	0.0013	-0.0049	-0.0013	0.0025

pixel correlation of the same position between the R, G and B components of the algorithm and comparative literature.

It can be seen from Figure 13 that the distribution of the three-channel same position correlation figure in the plaintext is extremely uneven, mainly distributed in the vicinity of three straight lines, indicating that the correlation between the plaintext and the three channels is very high. The intermediate ciphertext and ciphertext distribution are very uniform, indicating that the three-channel correlation is extremely low. It can be seen from Table 10 that the three-channel *CC* of the proposed algorithm is already very low, and the effective number that is the largest from 0 is also the last three digits of the decimal point, indicating that there is almost no relationship between them. At the same time, the correlation between the three positions of the three channels is better than that of the Reference [22], [23].

By analyzing histogram, χ^2 of the histogram, information entropy, and local Shannon entropy, the proposed algorithm can achieve good randomness in ciphertext distribution. The *CC* adjacent pixels and between R, G, B channels is also very

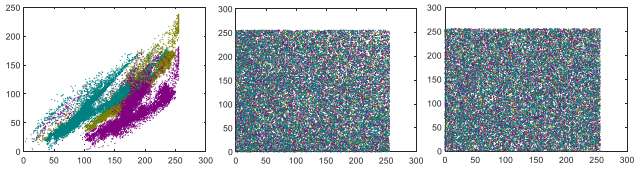


FIGURE 13. The same position correlation among R, G and B components.

low, so the algorithm can resist the statistical characteristic attack well.

E. ANTI-DIFFERENTIAL ATTACK ANALYSIS

A differential attack is an attacker attempting to find the relationship between two ciphertext images by modifying one pixel or one bit of the plaintext image [35]. In order to measure the influence of one pixel change on the ciphertext of the plaintext image, the pixel change rate (*NPCR*) and the unified average changing intensity (*UACI*) method are used to quantitatively calculate the resilience of the encryption algorithm to the differential attack. Its Equation is as follows [36]

$$NPCR = \frac{\sum_i \sum_j p(i,j)}{M \times N} \tag{31}$$

$$UACI = \frac{1}{M \times N}[\sum_i \sum_j \frac{|C1(i,j) - C2(i,j)|}{256}] \times 100\% \tag{32}$$

where $p(i,j) = 0$ when $C1(i,j) = C2(i,j)$, otherwise $p(i,j) = 1$. The ciphertexts $C1$ and $C2$ are respectively encrypted by using two plaintext images with only one pixel difference of 1, and the *NPCR* and *UACI* between the two ciphertexts are calculated as shown in Table 11. The expected values for NPCR and UACI for two random images of the uint8 type are 99.6094% and 33.4635% [35].

It can be seen from Table 11 that the proposed algorithm *NPCR* and *UACI* can reach 99.61 and 33.46, which is close to the ideal value, indicating that the sensitivity of the plaintext is high, and there is a strong avalanche effect between plaintext and ciphertext. It can be seen from Table 12 that the algorithm and the Reference [22], [21] have the same *NPCR* and *UACI*, which is more effective than others, so the algorithm can resist the differential attack well.

TABLE 11. NPCR and UACI between two ciphertexts.

Plaintext	Color *NPCR* (%)			Color *UACI* (%)			Gray *NPCR* (%)	Gray *UACI* (%)
	R	G	B	R	G	B		
Lena (256*256)	99.60	99.58	99.61	33.49	33.51	33.44	99.61	33.48
Teresa teng (256*256)	99.65	99.61	99.61	33.39	33.53	33.52	99.62	33.46
Baboon (512*512)	99.59	99.60	99.61	33.39	33.52	33.46	99.59	33.43
fingerprint (256*256)	—	—	—	—	—	—	99.62	33.47
White (256*256)	99.61	99.59	99.59	33.46	33.40	33.45	—	—
Black (256*256)	99.62	99.65	99.61	33.48	33.40	33.44	—	—
Mean		99.61			33.4587		99.61	33.46

TABLE 12. Compares the mean values of NPCR and UACI in different references.

Comparison item	Color			Gray			
	Our algorithm	Ref.[22]	Ref. [23]	Our algorithm	Ref. [19]	Ref. [20]	Ref. [21]
Average *NPCR*	99.61	99.61	99.60	99.61	99.66	99.85	99.61
Average *UACI*	33.46	33.46	33.46	33.46	33.50	33.42	33.46

TABLE 13. Differences between two encrypted images generated by slightly different keys.

Secret keys	*NPCR* (%)	*UACI* (%)
$x_{0,1}$ 10 ^{15}	99.60	33.44
μ_1 | 10 ^{15}	99.61	33.42
int $F_{0,1}$ 10 ^{15}	99.63	33.48
$k-1$	99.64	12.46
$x_{0,2}$ 10 ^{15}	99.63	33.43
μ_2 | 10 ^{15}	99.61	33.45
int $F_{0,2}$ 10 ^{15}	99.64	33.48

F. KEY SENSITIVITY ANALYSIS

The key sensitivity is that the ciphertext should be completely different when the key changes slightly during encryption. Similarly, the decryption results of the two keys should be different when decrypted. In the encryption process, because the encrypted images are all noise like images, this paper uses the *NPCR* and *UACI* of encrypted images before and after the key change to quantitatively analyze their differences. Figure 6 (I) is used as the plaintext, and the encrypted image is obtained when the key is slightly changed. The *NPCR* and *UACI* of the corresponding encrypted image and Figure 6 (k) are calculated as shown in Table 13. In the decryption process, we can easily use the naked eye to observe the decrypted image to determine whether the decryption is successful when the key is changed. Figure 6 (k) is used as the ciphertext, and the decryption image of the key change is obtained as shown in Figure 14, and calculate it the mean value of the absolute value of *CC* of the three channels with plaintext. And every time, only one parameter is changed and other parameters are constant.

It can be seen from Table 13 that when the chaotic key changes only 10^{-15}, the NPCR can reach 99.63% and the UACI can reach 33.45%, it is very close to the ideal values of 99.61% and 33.46% of the two random images. It can be seen from Figure 14 that when the key is slightly changed, the decrypted image is still a noise like image, and the CC between the decrypted image and the plaintext is also very close to 0, so the decrypted image can be considered as having no relationship with the plaintext at all. After the above analysis, it can be seen that the algorithm is sensitive to the key.

G. KEYSPACE ANALYSIS

The keys $key1(x_{0,1}, \mu_1, \text{int } F_{0,1}, k)$ and $key2(x_{0,2}, \mu_2, \text{int } F_{0,2})$ in this paper are the initial values and parameters for the 3D bit scrambling and diffusion operations, respectively. Among them, $\{0 \leq k \leq 8 \,|k \in N\}$, and each of the remaining chaotic keys uses double-precision data. It can be seen from IV. F that the chaotic key is very sensitive when changing 10^{-15}, so the 15 significant digits after the decimal point are reserved. It is calculated that the key space of the algorithm is at least 9×10^{90}. The encryption algorithm can also use another parameter M of chaos, the number of pre-iterations of the chaotic sequence and the SHA-256 value of the plaintext as keys, thereby further expanding the key. From the security point of view, the key space $\geq 2^{100} \approx 10^{30}$ can meet the higher security level [27], so the key space of this algorithm is safe for brute-force attack.

H. ANTI-SELECTIVE PLAINTEXT AND CIPHERTEXT ATTACK ANALYSIS

Since the choice of plaintext and ciphertext attacks is the most threatening to the encryption system, if the encryption system can resist the choice of plaintext and ciphertext attacks, it can resist other attacks against the encryption system [37]. Therefore, this paper uses the selected plaintext attack to further test the security of the system. Select plaintext attack, that is, the attacker already knows the encryption and decryption algorithms, and can arbitrarily select the plaintext and put it into the encryption system to obtain the corresponding ciphertext, and then analyze the key. The image obtained by adding the pixel value of one pixel in the Lena color image

TABLE 14. Comparison of time complexities of different algorithms.

Gray (s)				Color (s)		
Ref.[19]	Ref.[20]	Ref.[21]	Our algorithm	Ref.[22]	Ref.[23]	Our algorithm
0.886047	0.844556	1.336538	0.281012	1.301418	0.459120	0.733886

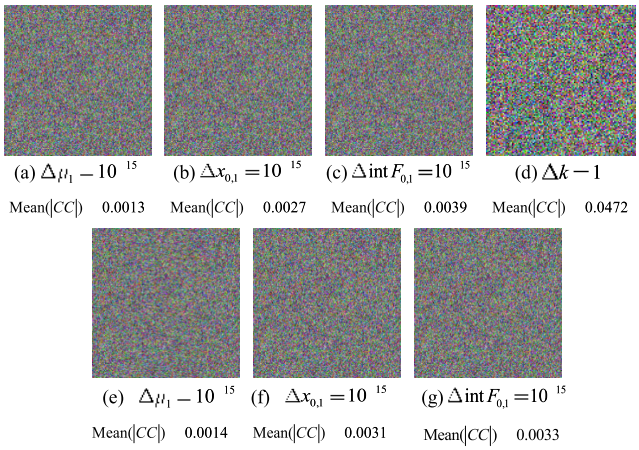


FIGURE 14. Decrypted image under error key and CC of it with plaintext.

FIGURE 15. Selecting a plaintext attack.

is used as an attack image, and a corresponding key stream is obtained. And decrypt it with the ciphertext of the plaintext of the Lena, the result is shown in Figure 15.

It can be seen from Figure 15 that when the attack image and the plaintext that should be obtained by the crack have only one pixel value difference of 1, the attack cannot be successful, which is enough to indicate that there is no possibility of successful cracking when the attack image gap is larger. This is mainly because the key is associated with the hash value SHA256 of the plaintext, which causes each pair of plaintext and ciphertext to have different keys, thereby generating different key streams, achieving ''one plaintext, one key'', The key stream of other plaintext and ciphertext pairs is used to decrypt different ciphertexts and lose the effect. The hash values of the attack image and the plaintext in Figure 15 are 34d336607f972f4dd755681328d2e0b666fa0708526a686ea-8ca9582475ea383 and 03d301011987ca5f194255ca81 67c-6ca1881272572f7ce9d08aaa589177e6254, respectiv-ely, and the key is changed from *key*1 (0.1305, 2.4062, 0.2320, 3), *key*2 (0.6258, 3.6191, 0.2469) to *key*1 (0.7555, 3.5879, 0.0797, 3), *key*2 (0.2508, 3.9805, 0.7664), so the key stream obtained from the attack image is simply not correct, so the cracked image is still a noise-like image.

I. COMPUTATION TIME ANALYSIS

The security of the encryption algorithm is important, and the encryption speed is also important. Table 14 lists the encryption time of each algorithm on the platform of chaotic time comparison.

It can be seen from Table 14 that the Reference [19]–[21] takes longer to encrypt the gray image, and even more than the encryption time of some algorithms for color images. This is mainly because in reference [19]–[21], pixels are scrambled first and then bits are scrambled. Compared with reference [22], [23] and the proposed algorithm, there is one more way to scramble pixel values, and the pixel scrambling is not repeated scrambling, and the bit scrambling is 8 bits scrambling of each pixel. These two kinds of scrambling cycle times are more. In Reference [22], [23], compared with the proposed algorithm, the number of cycles in bit scrambling is less, but the chaos they use is more complex, which makes the time of generating sequence longer. At the same time, in Reference [22], the process of transforming the chaotic generated sequence into the available pseudo-random sequence is too complex, which makes the encryption time longer than the proposed algorithm. From the above analysis, we can see that the efficiency of the proposed algorithm is very high.

V. CONCLUSION

This paper mainly improves in three aspects. Firstly, an L-F cascade chaos is designed, which solves the problem of blank windows in the uneven distribution of Logistic chaos, while retaining the rapidity of low-dimensional chaos. Moreover the number of the initial value and control parameters are increased, and the randomness of the sequence is enhanced. Secondly, a 3D bit encryption method is designed, which converts all pixels into binary sequences and reassembles them into 3D matrix, and then scrambles the higher bit-planes and the lower bit-planes by cyclic shifting respectively. This method is more sensitive than the existing scrambling method between 8 bits of one pixel. The algorithm has stronger anti-noise ability and better encryption effect. At the same time, we can adjust the randomness of the scrambled ciphertext and anti-noise ability of the algorithm by adjusting the number of higher bit-planes to meet more different encryption needs. Thirdly, the key is associated with the plaintext hash value so that the chaotic sequence can be adaptively changed with the plaintext, which increases the avalanche effect between plaintext and ciphertext and the ability to resist the attack of

selective plaintext. The experiments show that the algorithm can complete various types of images encrypt with high efficiency. The ciphertext distribution is uniform, the plaintext and the key sensitivity are strong. This algorithm can also resist common attack, and has high practicability and safety.

REFERENCES

[1] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation–based image copy–move forgery detection scheme," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 3, pp. 507–518, Mar. 2015.

[2] X.-Y. Wang, L.-X. Jiao, X.-B. Wang, H.-Y. Yang, and P.-P. Niu, "A new keypoint-based copy-move forgery detection for color image," *Appl. Intell.*, vol. 48, no. 10, pp. 3630–3652, Oct. 2018.

[3] Y. Li and J. Zhou, "Fast and effective image copy-move forgery detection via hierarchical feature point matching," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1307–1322, May 2019.

[4] K. X. Bing and S. M. Wei, "Adaptive threshold-based detection algorithm for image copy-move forgery," *Comput. Sci.*, vol. 38, no. 3, pp. 295–299, 2011.

[5] G. Ye and J. Zhou, "A block chaotic image encryption scheme based on self-adaptive modelling," *Appl. Soft Comput.*, vol. 22, pp. 351–357, Sep. 2014.

[6] 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.

[7] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons Fractals*, vol. 21, no. 3, pp. 749–761, Jul. 2004.

[8] Z. Congxu, L. I. Li, and C. Zhigang, "New image encryption algorithm based on combined multidimensional chaotic systems," *Comput. Eng.*, vol. 33, no. 2, pp. 142–144, 2007.

[9] Z.-L. Zhu, W. Zhang, K.-W. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Inf. Sci.*, vol. 181, no. 6, pp. 1171–1186, Mar. 2011.

[10] G. Zhang and Q. Liu, "A novel image encryption method based on total shuffling scheme," *Opt. Commun.*, vol. 284, no. 12, pp. 2775–2780, Jun. 2011.

[11] J. Wang and G. Jiang, "Cryptanalysis of a hyper-chaotic image encryption algorithm and its improved version," *Acta Phys. Sinica*, vol. 60, no. 6, pp. 868–870, 2011.

[12] X. Wang and G. He, "Cryptanalysis on a novel image encryption method based on total shuffling scheme," *Opt. Commun.*, vol. 284, no. 24, pp. 5804–5807, Dec. 2011.

[13] K.-H. Zhu and C.-X. Sun, "Cryptanalysis and improvement of a class of hyperchaos based image encryption algorithms," *Acta Phys. Sinica-Chin. Ed.*, vol. 61, no. 12, 2012, Art. no. 120503.

[14] G. Ye, "Image scrambling encryption algorithm of pixel bit based on chaos map," *Pattern Recognit. Lett.*, vol. 31, no. 5, pp. 347–354, Apr. 2010.

[15] L. Zhao, A. Adhikari, D. Xiao, and K. Sakurai, "On the security analysis of an image scrambling encryption of pixel bit and its improved scheme based on self-correlation encryption," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, no. 8, pp. 3303–3327, Aug. 2012.

[16] C. Li and K.-T. Lo, "Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal Process.*, vol. 91, no. 4, pp. 949–954, Apr. 2011.

[17] C. Li, D. Lin, and J. Lu, "Cryptanalyzing an image-scrambling encryption algorithm of pixel bits," *IEEE MultimediaMag.*, vol. 24, no. 3, pp. 64–71, Aug. 2017.

[18] H. Zhang and R. Cai, "Image encryption algorithm based on bit-plane scrambling and multiple chaotic systems combination," in *Proc. ICISS*, Guilin, China, Oct. 2010, pp. 113–117.

[19] X.-H. Deng, C.-L. Liao, C.-X. Zhu, and Z.-G. Chen, "Image encryption algorithms based on chaos through dual scrambling of pixel position and bit," *J. Commun.*, vol. 35, no. 3, pp. 216–223, 2014.

[20] X. Guo-Bo and W. Tian, "A novel hyperchaotic image encryption algorithm based on bit scrambling," *Microelectron. Comput.*, vol. 33, no. 7, pp. 28–32, 2016.

[21] S. Sun, "A novel hyperchaotic image encryption scheme based on dna encoding, pixel–level scrambling and bit–level scrambling," *IEEE Photon. J.*, vol. 10, no. 2, pp. 1–14, Apr. 2018.

[22] 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.

[23] X. Lv, X. Liao, and B. Yang, "Bit-level plane image encryption based on coupled map lattice with time-varying delay," *Mod. Phys. Lett. B*, vol. 32, no. 10, Apr. 2018, Art. no. 1850124.

[24] 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.

[25] A.-V. Diaconu, V. Ionescu, G. Iana, and J. M. Lopez-Guede, "A new bit-level permutation image encryption algorithm," in *Proc. Int. Conf. Commun. (COMM)*, Jun. 2016, pp. 411–416.

[26] Z. Shu-Qin, W. Wen-Hong, and S. Zhong-Gui, "Chosen plaintext attack on image encryption algorithm based on bit scrambling and hyperchaos," *Comput. Sci.*, vol. 33, no. 11, pp. 279–284, 2017.

[27] G. Yuan, X. Xin, and J. Shi-Wei, "Viryual optical image encrypyion method based on hybrid chaotic system," *Acta Photonica Sinica*, vol. 48, no. 7, 2019, Art. no. 0710002.

[28] Z. Wei, Y. Geng, and C. Lei, "An improved image encryption algorithm based on double random phase encoding and chaos," *Acta Optica Sinica*, vol. 34, no. 6, 2014, Art. no. 0607001.

[29] X.-L. Chai, Z.-H. Gan, K. Yuan, Y. Lu, and Y.-R. Chen, "An image encryption scheme based on three-dimensional Brownian motion and chaotic system," *Chin. Phys. B*, vol. 26, no. 2, Feb. 2017, Art. no. 020504.

[30] Y. Zhang and D. Xiao, "An image encryption scheme based on rotation matrix bit-level permutation and block diffusion," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 1, pp. 74–82, Jan. 2014.

[31] Z. G. Chen, D. Q. Liang, X. H. Deng, and Y. Zhang, "Performance analysis and improvement of logistic chaotic mapping," *J. Electron. Inf. Technol.*, vol. 38, no. 6, pp. 1547–1551, 2016.

[32] W.-J. Hsu, "Fibonacci cubes—A new interconnection topology," *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, no. 1, pp. 3–12, Jan. 1993.

[33] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, and P. Natarajan, "Local Shannon entropy measure with statistical tests for image randomness," *Inf. Sci.*, vol. 222, pp. 323–342, Feb. 2013.

[34] S. Amina and F. K. Mohamed, "An efficient and secure chaotic cipher algorithm for image content preservation," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 60, pp. 12–32, Jul. 2018.

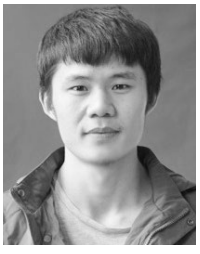
[35] S. K. Rajput and N. K. Nishchal, "Known-plaintext attack-based optical cryptosystem using phase-truncated Fresnel transform," *Appl. Opt.*, vol. 52, no. 4, p. 871, Feb. 2013.

[36] G. Liu, J. Li, and H. Liu, "Chaos-based color pathological image encryption scheme using one-time keys," *Comput. Biol. Med.*, vol. 45, pp. 111–117, Feb. 2014.

[37] Y. Li-Li, Y. Cao-Jin, Q. Jun-Jie, and N. Shou-Ping, "Asymmetric image encryption method based on gyrator transform and vector operation," *Acta Physica Sinica*, vol. 65, no. 21, 2016, Art. no. 214203.

YUAN GUO received the B.S. degree in automation from Qiqihar University, Qiqihar, China, in 1997, and the M.S. and Ph.D. degrees in electrical engineering from Yanshan University, Qinhuangdao, China, in 2004 and 2008, respectively.

She was a Visiting Scholar with Johns Hopkins University, Baltimore, MD, USA, from 2012 to 2013. She is currently a Professor of computer science and technology with Qiqihar University. Her current research interests include photoelectric detection, optical image encryption, sensor technology, and image processing.



SHIWEI JING received the B.S. degree in electrical engineering and automation from Qiqihar University, Qiqihar, China, in 2018, where he is currently pursuing the M.S. degree. His current research interests include optical image encryption and image processing.

YANYAN ZHOU received the B.S. degree in electrical engineering and automation from Shandong Jiaotong University, Jinan, China, in 2016. She is currently pursuing the M.S. degree with Qiqihar University. Her current research interest is image processing.

XIN XU received the B.S. degree in electronic science and technology from the Liren College, Yanshan University, Qinhuangdao, China, in 2017. She is currently pursuing the M.S. degree with Qiqihar University. Her current research interest is image processing.

LIANSUO WEI received the B.S. degree in mathematics and the M.S. degree in computer engineering and application electrical engineering from Qiqihar University, Qiqihar, China, in 2003 and 2010, respectively. He is currently pursuing the Ph.D. degree with Harbin Engineering University. His current research interests include artificial intelligence and pattern recognition, sensor technology, underwater sensor networks, and information processing and simulation.