Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 3965281, 22 pages
https://doi.org/10.1155/2020/3965281

Research Article

Chaotic Color Image Encryption Scheme Using Deoxyribonucleic Acid (DNA) Coding Calculations and Arithmetic over the Galois Field

Lilian Huang, Shiming Wang, Jianhong Xiang, and Yi Sun

College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Jianhong Xiang; xiangjianhong@hrbeu.edu.cn

Received 20 September 2019; Revised 27 November 2019; Accepted 31 January 2020; Published 9 March 2020

Guest Editor: Marco Perez-Cisneros

This paper proposes a chaotic color image encryption scheme based on DNA-coding calculations and arithmetic over the Galois field. Firstly, three modified one-dimensional (1D) chaotic maps with larger key space and better chaotic characteristics are presented. The experimental results show that their chaotic intervals are not only expanded to $(0, 15]$, but their average largest Lyapunov Exponent reaches 10. They are utilized as initial keys. Secondly, DNA coding and calculations are applied in order to add more permutation of the cryptosystem. Ultimately, the numeration over the Galois field ensures the effect for the diffusion of pixels. The simulation analysis shows that the encryption scheme proposed in this paper has good encryption effect, and the numerical results verify that it has higher security than some of the latest cryptosystems.

1. Introduction

Information security is an important issue in information communication nowadays. With the advancement of information technology, plenty of digital contents are stored and transmitted in various forms. Therefore, it becomes more and more significant to improve the security of them. As the main information carrier, digital images play an important role in the medical and military fields. Hence, the safety of image transmission has received extensive attention. Nevertheless, since the images have large amounts of data, high redundancy and strong correlation of adjacent pixels, conventional encryption technologies such as DES, AES, and RSA fail to satisfy the security of encryption.

Chaotic systems are ergodic, dispersive, and highly sensitive to initial conditions [1–4]. They have a lot of similarities with cryptography. As a consequence, the chaos-based encryption methods have become a main branch of the cryptosystem. Fridrich first applied chaotic maps to image encryption algorithms, who used two-dimensional (2D) Baker map and Cat map for pixel position transformation in 1997 [5]. After that, many researchers encrypted images based on 1D and multidimensional (MD) chaotic maps [6–10]. Among them, the MD chaotic maps are widely used in image encryption due to their relatively intricate structures and parameters [11–14]. However, these traits increase the complexity of computation and the difficulty of implementation. In comparison, although the 1D chaotic maps have uneven distribution and discontinuous range, they have simpler structures. Thus, they are more convenient to be handled and implemented. In recent years, some efforts have been devoted to addressing the weaknesses of 1D chaotic systems and proposing encryption schemes. Zhou et al. presented a new system structure and encrypted images with random pixel insertion [15]. However, Dhall et al. pointed out that this cryptosystem can be broken by the differential attack [16]. Hua et al. introduced a Sine-Logistic modulation map to efficiently change the image pixel positions in [17]. Pak and Huang also proposed a new 1D chaotic system to determine pixels permutation and diffusion positions [18]. But the algorithms in [17, 18] are vulnerable to chosen plaintext attack. Hence, we propose three improved 1D chaotic maps with better dynamic complexity for our encryption scheme. The simulation results exhibit

that the new 1D chaotic maps have larger chaotic ranges and more uniform output sequences, which makes them more suitable for image encryption.

DNA-based image encryption methods mainly apply the principle of DNA cryptography. Due to huge parallelism, inherent information density, and fairly low power consumption, DNA-based image encryption methods have been rapidly developed. In [19], Zhang et al. proposed an image encryption scheme with DNA sequence addition operation and two Logistic maps. However, the Logistic map has been declared that it does not have outstanding randomness to achieve the desired encryption effect [15]. Hence, Li et al. changed the location of pixels and pixel values with the Arnold map before DNA encoding [20]. Since the key streams are independent of the plain images, Yong proved that this scheme can be cracked by the chosen plaintext attack and known plaintext attack [21]. The researchers also considered the application of chaotic systems with excellent chaotic dynamic properties to improve the robustness of the cryptosystems. In [22], Zhen et al. introduced an encryption method based on spatiotemporal chaotic systems and Logistic map. Moreover, the method also utilized DNA coding to promote the efficiency of image confusion and diffusion. Although the authors have claimed that the cryptosystem in [22] can resist a variety of attacks, Xin et al. still found two flaws that make the encryption fail under chosen plaintext attack [23].

Motivated by the abovementioned discussions, we propose a new chaotic color image encryption scheme using DNA coding calculations and arithmetic over the Galois field. Particularly, the 1D chaotic maps applied in this paper are derived from the classical Logistic map, Sine map, and Chebyshev map. Besides, the contributions and innovations of this article are summarized as follows. (1) We propose improved 1D chaotic maps that are more appropriate for image encryption. Not only do they have better chaotic properties but they are also easier to be operated on hardware and software. (2) We design the calculation modes of the encoded DNA matrixes to be randomly decided by the chaotic sequences which are updated by plain images. In this case, the cryptosystem is more robust. (3) In order to prevent the danger of being cracked by utilizing DNA operations only, we add multiplication arithmetic over the Galois field GF(17) to our scheme. Moreover, the generation of the lookup table will not increase the time consumption.

The arrangement of this paper is as follows. Section 2 introduces the basic theories involved in this paper. Section 3 displays the improved 1D chaotic maps with their performance analysis. Section 4 proposes our image encryption and decryption schemes. Section 5 gives the experimental simulation results and security analysis. The Section 6 draws the conclusion.

2. Related Work

2.1. DNA Coding and Calculation. DNA coding is a concept derived from biology. A DNA sequence consists of four nucleotides. They are adenine (A), guanine (G), cytosine (C), and thymine (T). According to the principle of complementary base pairing, adenine (A) pairs with thymine (T) and cytosine (C) pairs with guanine (G) [24]. As shown in Table 1, there are eight DNA coding methods that satisfy the pairing rules [25]. In the binary system, 0 and 1 are also complementary. Thus, each 8-bit pixel value can be decomposed into four "2-bit" values, and the four values can be encoded with a certain coding rule to obtain a DNA sequence. For instance, if the value of a pixel is 177, it is "10110001" in binary encoding. One can get the DNA sequence "CTAG" by the DNA coding rule 2 in Table 1. And using disparate decoding rules will obtain diverse pixel values. Hence, when the sequence "CTAG" is decoded by the DNA decoding rule 5 in Table 1, the binary result "00100111" is obtained, and the corresponding decimal number is 39.

In addition to encoding pixel values, DNA sequences can also perform algebraic calculations. Because the operation modes of DNA are based on traditional arithmetic operations in binary, the eight DNA coding rules correspond to eight kinds of DNA addition, DNA subtraction, and DNA exclusive or (XOR) rules [26]. If we arbitrarily select the DNA coding rule 1 in Table 1, then the corresponding DNA addition, subtraction, and XOR modes are shown in Tables 2–4.

2.2. Multiplication over the Galois Field. The French mathematician Galois invented the Galois field, and the operations of addition, subtraction, multiplication, and division over the Galois field can be performed. Compared with the addition and subtraction over the Galois field, the multiplication and division are more complicated. That is, the diffusion effect on pixels is more prominent. In order to speed up the calculation, we need to construct a lookup table in advance by

$$\left((0\colon 16)^{T} \times (0\colon 16)\right) \bmod 17, \tag{1}$$

where $(\cdot)^{T}$ denotes transposition of the sequence. As for the GF(17) multiplication, a 8-bit pixel value is divided into the upper four bits and the lower four bits whose value ranges are $[0, 15]$. The bits are further converted to $[1, 16]$ when the lookup table is established, and the results are exhibited in Table 5.

3. Modification of 1D Chaotic Maps

With only one variable and simple structures, 1D chaotic maps have low implementation costs. Hence, they are suitable for efficient image encryption. Nevertheless, existing 1D chaotic maps still have some defects. In this section, we will introduce and analyze three conventional 1D chaotic maps, and further present three modified 1D chaotic maps.

3.1. Three Classical 1D Chaotic Maps. The first classical chaotic map is the Logistic map, which is also called the insect mouth model. Its definition is as follows:

$$X_{k+1} = F_L(\mu, X_k) = \mu X_k (1 - X_k), \tag{2}$$

Table 1: 8 DNA coding and decoding rules.

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

Table 2: DNA addition.

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

Table 3: DNA subtraction.

−	A	T	G	C
A	A	G	T	C
T	T	A	C	G
G	G	C	A	T
C	C	T	G	A

Table 4: DNA XOR operation.

⊕	A	T	G	C
A	A	T	G	C
T	T	A	C	G
G	G	C	A	T
C	C	G	T	A

Table 5: Multiplication lookup table on the GF(17).

×	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	0	2	4	6	8	10	12	14	16	1	3	5	7	9	11	13	15
3	0	3	6	9	12	15	1	4	7	10	13	16	2	5	8	11	14
4	0	4	8	12	16	3	7	11	15	2	6	10	14	1	5	9	13
5	0	5	10	15	3	8	13	1	6	11	16	4	9	14	2	7	12
6	0	6	12	1	7	13	2	8	14	3	9	15	4	10	16	5	11
7	0	7	14	4	11	1	8	15	5	12	2	9	16	6	13	3	10
8	0	8	16	7	15	6	14	5	13	4	12	3	11	2	10	1	9
9	0	9	1	10	2	11	3	12	4	13	5	14	6	15	7	16	8
10	0	10	3	13	6	16	9	2	12	5	15	8	1	11	4	14	7
11	0	11	5	16	10	4	15	9	3	14	8	2	13	7	1	12	6
12	0	12	7	2	14	9	4	16	11	6	1	13	8	3	15	10	5
13	0	13	9	5	1	14	10	6	2	15	11	7	3	16	12	8	4
14	0	14	11	8	5	2	16	13	10	7	4	1	15	12	9	6	3
15	0	15	13	11	9	7	5	3	1	16	14	12	10	8	6	4	2
16	0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

where parameter $\mu \in (0, 4]$, and X_k is the output sequence. The bifurcation diagram and Lyapunov Exponent of the Logistic map are shown in Figures 1(a) and 2(a), respectively. It can be seen from the bifurcation diagram that when the parameter is 3, the Logistic map appears 2 bifurcations from the steady-state solution, and it does not enter chaos until μ is close to 4. The chaotic region of the Logistic map is narrow and a blank window will appear. Only when the Lyapunov exponent exhibits a positive state, the chaotic map has excellent chaotic property. However, the Lyapunov exponents of the Logistic map are negative when $\mu < 3.57$. Figure 3(a) shows the chaotic

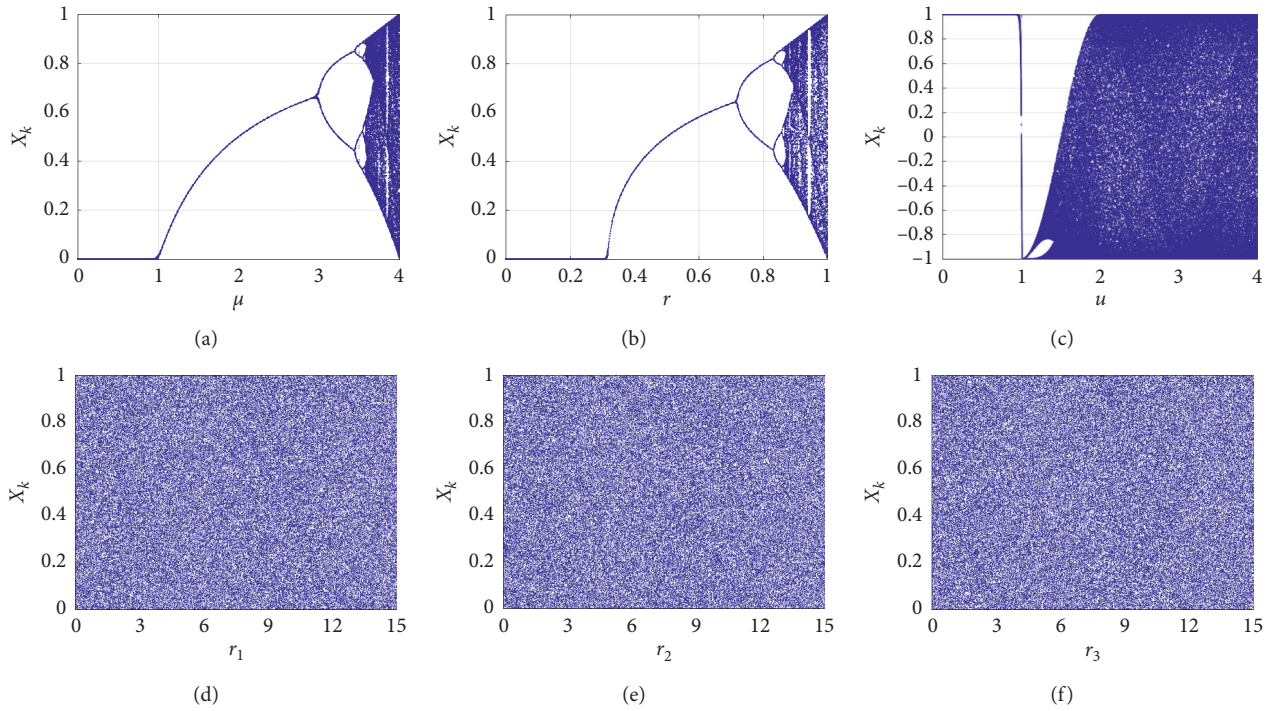


Figure 1: The bifurcation diagrams of the (a) Logistic map; (b) Sine map; (c) Chebyshev map; (d) SLM; (e) CLM; (f) SCM.

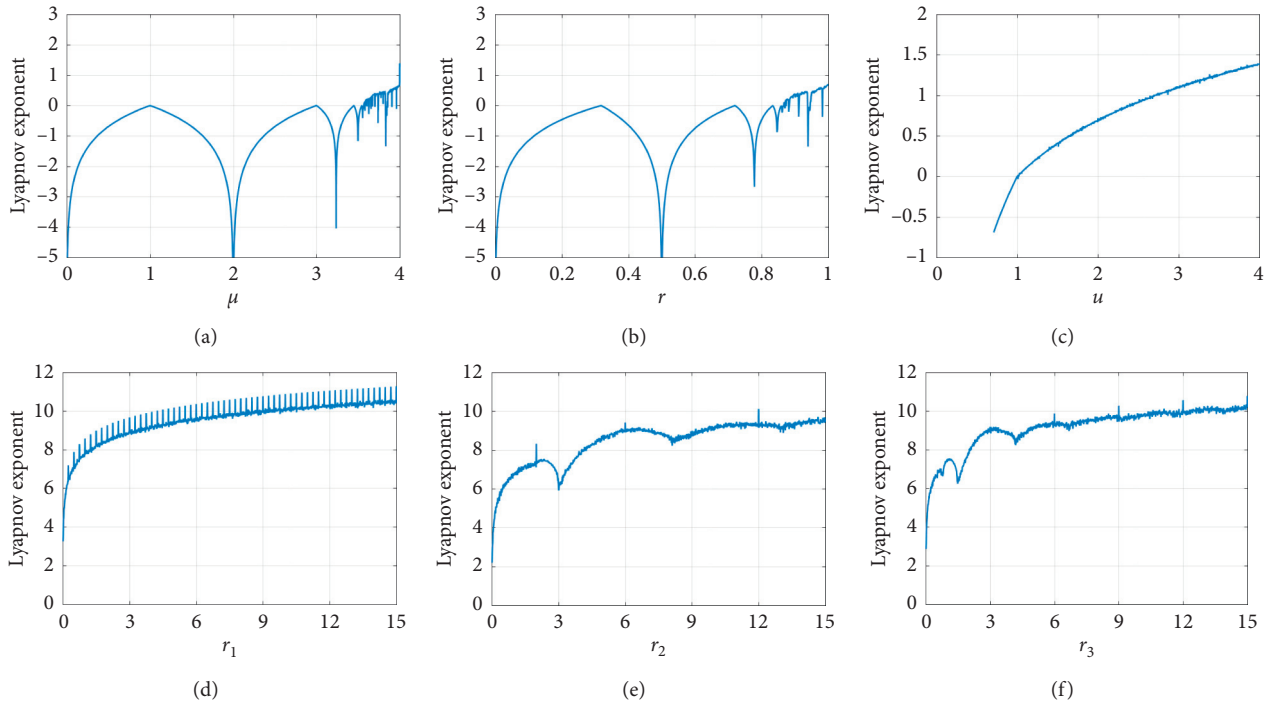


Figure 2: The Lyapunov Exponents of the (a) Logistic map; (b) Sine map; (c) Chebyshev map; (d) SLM; (e) CLM; (f) SCM.

sequence distribution with the initial value of 0.1, the bifurcation parameter of 4, and the iterations of 10000. Most values are close to 0 and 1. This indicates that the distribution of the sequence generated by the Logistic map is uneven. Hence, the application range of the Logistic map is tiny.

The Sine map has similar chaotic properties with the Logistic map [17], which is described by

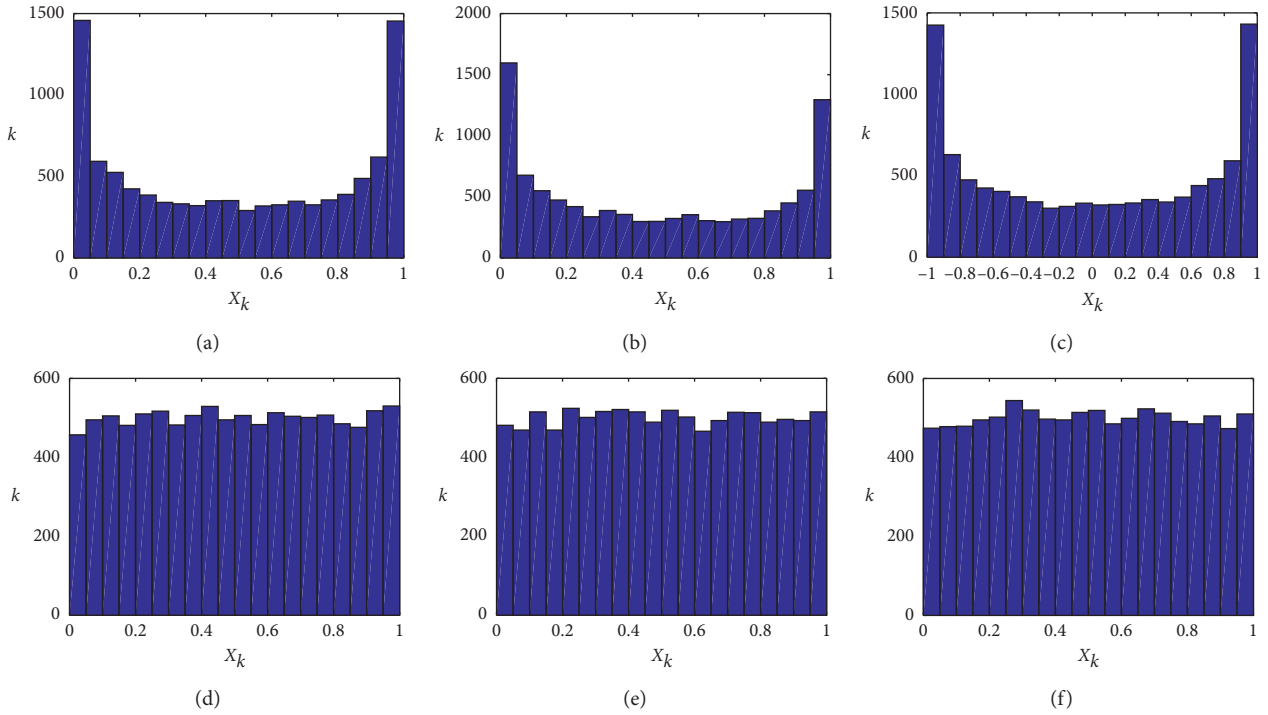


Figure 3: The distribution of the chaotic sequences for the (a) Logistic map; (b) Sine map; (c) Chebyshev map; (d) SLM; (e) CLM; (f) SCM.

$$X_{k+1} = F_S(r, X_k) = r \sin(\pi \times X_k), \tag{3}$$

where parameter $r \in (0, 1]$. The bifurcation analysis of the Sine map is exhibited in Figure 1(b), and the map has chaotic characteristic when r is close to 1. It can be seen in Figure 2(b) that a portion of the Lyapunov exponents for the Sine map are positive and others of them are negative. Figure 3(b) shows that the chaotic sequence distribution of the Sine map is similar to the Logistic map when the bifurcation parameter is set to 1 and the other parameters are unchanged. It is also not uniform. Thus, there are a number of security risks when using the Sine map for encryption.

The Chebyshev map is also a traditional 1D chaotic map [27], and it can be expressed as follows:

$$X_{k+1} = F_C(u, X_k) = \cos(u \times \arccos X_k), \tag{4}$$

where parameter $u \in N$. It can be seen from the bifurcation diagram in Figure 1(c) that the map does not have chaos property when $u < 1$, and it has obvious blank areas within $[1, 2]$. Only when u is greater than 2, the map actually enters a chaotic state. Figure 2(c) displays that a portion of the Lyapunov exponents for the Chebyshev map are negative. Figure 3(c) shows the sequence distribution of the same parameters set as the Logistic map, and a great many values are close to 0 and 1. It also proves that the application range of the Chebyshev map is tiny and limited.

3.2. Three Modified 1D Chaotic Maps. In this section, the 1D chaotic maps mentioned above are combined to put forward three modified 1D chaotic maps. The purpose is to rectify the shortcomings of the classical 1D chaotic maps analyzed in the previous section. It can be found that the novel 1D chaotic maps have better chaotic properties than the original chaotic maps.

The first map is the SLM (Sine-Logistic map), which can be defined by the following formula:

$$\begin{aligned} X_{k+1} = {} & f(n) \times (F_S(r_1, X_k) + F_L(2r_1, X_k)) \\ & - \lfloor (f(n) \times (F_S(r_1, X_k)) + F_L(2r_1, X_k)) \rfloor, \end{aligned} \tag{5}$$

where $f(n) = 2^n$ with $10 \le n \le 20$, which is used to balance the entire function. The symbol $\lfloor \cdot \rfloor$ represents the integer function that make the element return the nearest integer towards minus infinity. It is utilized to control the chaotic sequence within the range of $(0, 1]$. After the improvement, the range of bifurcation parameter r_1 is increased to $(0, 15]$, and x_0 is the initial value of the output sequence. It can be seen from the bifurcation diagram in Figure 1(d) that there is no obvious blank area in the entire chaotic region. In other words, the improved SLM expands the original chaotic range. Moreover, Figure 2(d) displays that the Lyapunov exponents of the SLM are all positive within $(0, 15]$, and Figure 3(d) shows that its chaotic sequence is uniformly distributed when the parameter set is the same as the Logistic map.

The second map is the CLM (Chebyshev-Logistic map). Its formula can be described as

$$\begin{aligned} X_{k+1} = {} & f(n) \times (F_C(r_2, X_k) + F_L(2r_2, X_k)) \\ & - \lfloor (f(n) \times F_C(r_2, X_k) + F_L(2r_2, X_k)) \rfloor, \end{aligned} \tag{6}$$

where $f(n)$ has the same effect as it in the SLM and control parameter $r_2 \in (0, 15]$. From the bifurcation diagram in

Figure 1(e) and Lyapunov exponent in Figure 2(e) of the CLM, it can be found that the CLM has superior chaotic behavior, which is similar to the SLM. Besides, Figure 3(e) shows that the distribution of its chaotic sequence is semblable to the SLM.

The third map is the SCM (Sine-Chebyshev map). The combination equation is as follows:

$$X_{k+1} = f(n) \times \left(F_S(r_3, X_k) + F_C(2r_3, X_k)\right) - \lfloor \left(f(n) \times F_S(r_3, X_k) + F_C(2r_3, X_k)\right) \rfloor, \tag{7}$$

where $f(n)$ is the same as it in equation (6) and control parameter $r_3 \in (0, 15]$. The bifurcation diagram and Lyapunov exponent of the SCM are revealed in Figures 1(f) and 2(f), respectively. Moreover, the distribution of the output sequence is displayed in Figure 3(f). Its chaos property is similar with the SLM and CLM, which is also improved a lot.

Table 6 lists the comparison of characteristics between our maps and other improved 1D chaotic maps. It can be seen that the maps proposed in this paper have better chaotic behaviors.

4. The Proposed Image Encryption and Decryption Scheme

4.1. Encryption Scheme

Input: color plain image P of size $m \times n$, the security keys which are composed of the bifurcation parameters r_1, r_2, and r_3 and three initial values X_{10}, X_{20}, and X_{30} of the SLM, CLM, and SCM.

Output: the color cipher image with the same size.

Step 1: decompose the image P into three matrixes R, G, and B. Then, update the three initial values with bit-planes recombination according to the following formula:

$$\begin{cases} X'_{10} = X_{10} + \dfrac{\sum R_{\text{odd}}(x, y)}{255 \times n^2}, \\ X'_{20} = X_{20} + \dfrac{\sum G_{\text{odd}}(x, y)}{5 \times m^2 \times n^2}, \\ X'_{30} = X_{30} + \dfrac{\sum B_{\text{odd}}(x, y)}{5 \times m^2 \times n^2}, \end{cases} \tag{8}$$

$$\begin{cases} \left(X_1^1(i)\right)_{i=1}^{m \times n} = \left(X_1^1(1), X_1^1(2), \ldots, X_1^1(m \times n)\right), \\ \left(X_2^1(i)\right)_{i=1}^{m \times n} = \left(X_2^1(1), X_2^1(2), \ldots, X_2^1(m \times n)\right), \\ \left(X_3^1(i)\right)_{i=1}^{m \times n} = \left(X_3^1(1), X_3^1(2), \ldots, X_3^1(m \times n)\right). \\ \left(X_1^2(i)\right)_{i=m \times n+1}^{2 \times m \times n} = \left(X_1^2(m \times n + 1), X_1^2(m \times n + 2), \ldots, X_1^2(2 \times m \times n)\right), \\ \left(X_2^2(i)\right)_{i=m \times n+1}^{2 \times m \times n} = \left(X_2^2(m \times n + 1), X_2^2(m \times n + 2), \ldots, X_2^2(2 \times m \times n)\right), \\ \left(X_3^2(i)\right)_{i=m \times n+1}^{2 \times m \times n} = \left(X_3^2(m \times n + 1), X_3^2(m \times n + 2), \ldots, X_3^2(2 \times m \times n)\right). \end{cases} \tag{9}$$

$$\begin{cases} \widetilde{X}_1^1(i) = \left(\lfloor X_1^1(i) + 50 \rfloor \times 10^{12}\right) \bmod (m \times n) + 1, \\ \widetilde{X}_2^1(i) = \left(\lfloor X_2^1(i) + 100 \rfloor \times 10^{12}\right) \bmod (m \times n) + 1, \\ \widetilde{X}_3^1(i) = \left(\lfloor X_3^1(i) + 500 \rfloor \times 10^{12}\right) \bmod (m \times n) + 1. \end{cases} \tag{10}$$

$$\begin{cases} V_1(i) = V_1\left(\widetilde{X}_1^1(i)\right), \\ V_2(i) = V_2\left(\widetilde{X}_2^1(i)\right), \\ V_3(i) = V_3\left(\widetilde{X}_3^1(i)\right). \end{cases} \tag{11}$$

$$\begin{cases} \widehat{X}_1^1(i) = \text{fix}\left(X_1^1(i) \times 10^4\right) \bmod 256, \\ \widehat{X}_2^1(i) = \text{fix}\left(X_2^1(i) \times 10^4\right) \bmod 256, \\ \widehat{X}_3^1(i) = \text{fix}\left(X_3^1(i) \times 10^4\right) \bmod 256. \end{cases} \tag{12}$$

Table 6: Chaotic characters of improved 1D chaotic maps.

Chaotic maps	Range of bifurcation parameters	Average maximum Lyapunov exponent
The proposed chaotic maps	(0, 15]	10
Ref. [15]	(0, 4]	0.7
Ref. [18]	(0, 10]	2
Ref. [28]	[−4, 4]	1

$$\begin{cases} \widehat{X}_1^2(i) = \mathrm{fix}\left(X_1^1(i) \times 2^{16}\right) \bmod 256, \\ \widehat{X}_2^2(i) = \mathrm{fix}\left(X_2^1(i) \times 2^{16}\right) \bmod 256, \\ \widehat{X}_3^2(i) = \mathrm{fix}\left(X_3^1(i) \times 2^{16}\right) \bmod 256. \end{cases} \tag{13}$$

$$\overline{X}_1^1(i) = \lceil X_1^1(i) \times 10^4 \rceil \bmod 3, \tag{14}$$

$$E_{DR,DG,DB} = \begin{cases} C_{DR,DG,DB} + M_{D1,D2,D3}, & \text{if } \overline{X}_1(i) = 0, \\ C_{DR,DG,DB} - M_{D1,D2,D3}, & \text{if } \overline{X}_1(i) = 1, \\ C_{DR,DG,DB} \oplus M_{D1,D2,D3}, & \text{if } \overline{X}_1(i) = 2. \end{cases} \tag{15}$$

$$\widehat{X}_2^1(i) = \lfloor X_2^1(i) \times 10^4 \rfloor \bmod 8 + 1. \tag{16}$$

$$\begin{cases} C_{ER}(i) = C_{ER}(i-1) \times \widehat{X}_1^1(i) \times C2_R(i), \\ C_{EG}(i) = C_{EG}(i-1) \times \widehat{X}_2^1(i) \times C2_G(i), \\ C_{EB}(i) = C_{EB}(i-1) \times \widehat{X}_3^1(i) \times C2_B(i). \end{cases} \tag{17}$$

$$\begin{cases} C_R(i) = C_R(i-1) \times \widehat{X}_1^2(i) \times C_{ER}(i), \\ C_G(i) = C_G(i-1) \times \widehat{X}_2^2(i) \times C_{EG}(i), \\ C_B(i) = C_B(i-1) \times \widehat{X}_3^2(i) \times C_{EB}(i). \end{cases} \tag{18}$$

where $\sum (R, G, B)_{\mathrm{odd}}$ denotes sum of the odd bit-planes about components R, G, and B.

Step 2: iterate the SLM, CLM, and SCM $(2 \times m \times n + 500)$ times with X'_{10}, X'_{20}, and X'_{30}, respectively. Discard the first 500 elements to eliminate the effects of transient processes. Then, two sets of new sequences are formed, which can be expressed as follows:

Step 3: convert the components R, G, and B into three 1D vectors V_1, V_2, and V_3, and then perform pixel-level scrambling on them according to equations (10) and (11):

Then, restore the three vectors to the matrixes $C1_R$, $C1_G$, and $C1_B$.

Step 4: generate two new sets of sequences by equations (12) and (13) After that, reshape the sequences in equation (12) to the matrixes M_1, M_2, and M_3 of size $m \times n$:

where $\mathrm{fix}(\cdot)$ indicates the function rounds the element to the nearest integer towards minus infinity.

Step 5: fill each matrix into a square with zero elements and separate each matrix into parts of size $s \times s$, respectively. Arbitrarily, select the DNA coding rule 1 in Table 1 to encode each part of $C1_R$, $C1_G$, $C1_B$, M_1, M_2, and M_3 from steps 3 and 4. Then, six DNA matrixes named $C1_{DR}$, $C1_{DG}$, $C1_{DB}$, M_{D1}, M_{D2}, and M_{D3} are obtained. After this, utilize equation (14) to execute DNA calculation on these matrixes:

where $\lceil \cdot \rceil$ denotes that the element returns the smallest integer in the infinite direction, and the elements in $(\overline{X}_1^1(i))_{i=1}^{m \times n}$ are 0, 1, and 2. Hence, the corresponding DNA calculations are addition, subtraction, and XOR operations, which can be defined as follows:

Step 6: utilize equation (16) to determine the decoding rules of equation (15). The DNA matrixes are decoded to binary matrixes and further transformed into decimal numbers to get three encrypted components $C2_R$, $C2_G$, and $C2_B$:

Step 7: implement multiplication on the GF(17) with Table 5 according to equations (17) and (18):

It should be noted that the matrixes need to be converted into the sequences for multiplication, and the three components C_R, C_G, and C_B are merged to form

the final encrypted image C. The flowchart of our encryption scheme is revealed in Figure 4.

4.2. Decryption Scheme. The decryption procedure is the inverse process of the encryption procedure. Thus, the steps presented in the previous section ought to proceed in the reverse order. However, the only distinction is that step 5 needs to apply equation (19) to determine the DNA calculation mode when decrypting:

$$E_{DR,DG,DB} = \begin{cases} C_{DR,DG,DB} - M_{D1,D2,D3}, & \text{if } \overline{X}_1(i) = 0, \\ C_{DR,DG,DB} + M_{D1,D2,D3}, & \text{if } \overline{X}_1(i) = 1, \\ C_{DR,DG,DB} \oplus M_{D1,D2,D3}, & \text{if } \overline{X}_1(i) = 2. \end{cases} \tag{19}$$

5. Experimental Results and Security Analysis

The experiment is implemented by MATLAB R2016b on a PC with an Intel Core i5, 3.4 GHz CPU, 8 GB memory, and the encryption results for three color images "Female", "Peppers," and "Mandrill" of size 256×256 are displayed in Figure 5. The control parameters and initial values are set as follows: $r_1 = 14.99120306595001$, $r_2 = 14.985633002586235$, $r_3 = 14.978965662236302$, $X_{10} = 0.654321563325991$, $X_{20} = 0.563214562356231$, and $X_{30} = 0.456326656565231$. We specify these six elements as the key set K_0. Visually, the scheme designed in this paper has good performance. The ciphered images are noise-like ones, and the decrypted images are almost identical with the plain images.

5.1. Differential Attack Analysis. In general, the intensity of sensitivity to the plain image can determine the ability of the algorithm to resist differential attack. The measured indicators are the number of pixels change rate (NPCR) and the unified average changing intensity (UACI), which can be defined as follows:

$$\begin{aligned} \mathrm{NPCR}_{R,G,B} &= \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} E_{R,G,B}(i,j) \times 100\%, \\ \mathrm{UACI}_{R,G,B} &= \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{\left| C_{R,G,B}(i,j) - C'_{R,G,B}(i,j) \right|}{255} \times 100\%, \end{aligned} \tag{20}$$

where M and N are the width and height of the image and C and $C\prime$ are two cipher images whose plain images have only one different pixel. Besides, $E_{R,G,B}(i,j)$ is used to determine the distinction between two ciphered images, which can be calculated by

$$E_{R,G,B}(i,j) = \begin{cases} 0, & C_{R,G,B}(i,j) = C'_{R,G,B}(i,j), \\ 1, & C_{R,G,B}(i,j) \neq C'_{R,G,B}(i,j). \end{cases} \tag{21}$$

Nevertheless, Zhang [31] pointed out that these two indexes cannot precisely measure the difference between two images. He proposed block average changing intensity (BACI) to quantitatively analyze the antidifferential attack characteristic. It is a method of block calculation about subtraction images, and each small block can be defined by

$$B_i = \begin{bmatrix} b_{i1} & b_{i2} \\ b_{i3} & b_{i4} \end{bmatrix}. \tag{22}$$

Moreover, the mean for the absolute values of differences between arbitrary two elements can be defined as

$$\begin{aligned} m_i = \frac{1}{6}\big(&\left|b_{i1} - b_{i2}\right| + \left|b_{i1} - b_{i3}\right| + \left|b_{i1} - b_{i4}\right| + \left|b_{i2} - b_{i3}\right| \\ &+ \left|b_{i2} - b_{i4}\right| + \left|b_{i3} - b_{i4}\right|\big). \end{aligned} \tag{23}$$

Then, the BACI can be calculated by

$$\mathrm{BACI}_{R,G,B} = \frac{1}{(M-1)(N-1)} \sum_{i=1}^{(M-1)(N-1)} \frac{m_i}{255}. \tag{24}$$

Thus, the NPCR, UACI, and BACI are used together to analyze the sensitive property of our encryption scheme, and the results are revealed in Table 7. It can be found that the measured values of six different color images are very close to the theoretical values 99.6094%, 33.4635%, and 26.7712%, respectively. In particular, Table 7 also compares the NPCR and UACI scores on the Lena, Mandrill, and Peppers images using our scheme with algorithms in [29, 30]. This further demonstrates that our cryptosystem has excellent ability to resist differential attack.

5.2. Key Space. As mentioned earlier in this section, the secret keys for our scheme consist of three control parameters (r_1, r_2, r_3) and three initial values (X_{10}, X_{20}, X_{30}), which are all double-precision real numbers. In this paper, we use the 64-bit double-precision format in [32] to calculate our key space, and the total key space of our scheme can reach $(10^{15})^6 = 10^{90} \approx 2^{299}$. In addition, Table 8 exhibits the comparison of key space between our scheme and other chaos-based encryption algorithms. It can be seen that the key space of our method is large enough to resist brute-force attack.

5.3. Key Sensitivity. The key sensitivity refers to the degree of variation for the corresponding ciphered image when the initial key alters slightly. Make a minor change to X_{10} while keeping other keys unchanged. For instance, $X_{10}' = X_{10} + 10^{-15}$. This produces a new key set K_1. It is applied to encrypt the plain images in Figure 5(a), and the ciphered images in Figure 6(a) are displayed. The pixel-by-pixel differences between Figures 5(b) and 6(a) are revealed in Figure 6(b), which proves that their encrypted images are completely different. Moreover, K_0 and K_1 are utilized to decrypt the ciphered images in Figure 5(b), respectively. The decrypted images in Figure 6(c) can be acquired with the correct key set K_0, and a minor change to the security keys will cause the failure of decryption, as shown in Figure 6(d).

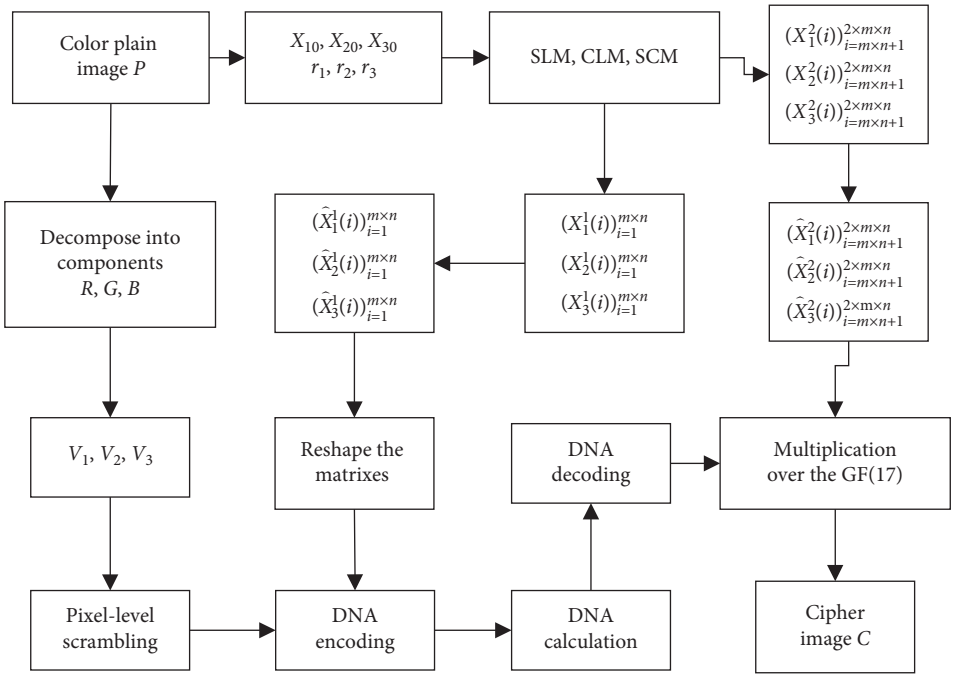
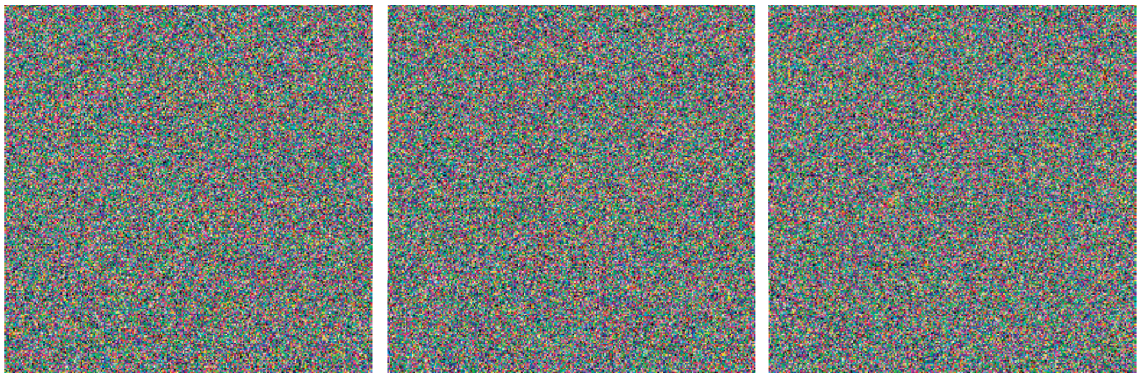


Figure 4: Flowchart of the encryption scheme.

(a)



(b)

Figure 5: Continued.

(c)

Figure 5: Encryption results: (a) original images; (b) encrypted images; (c) decrypted images.

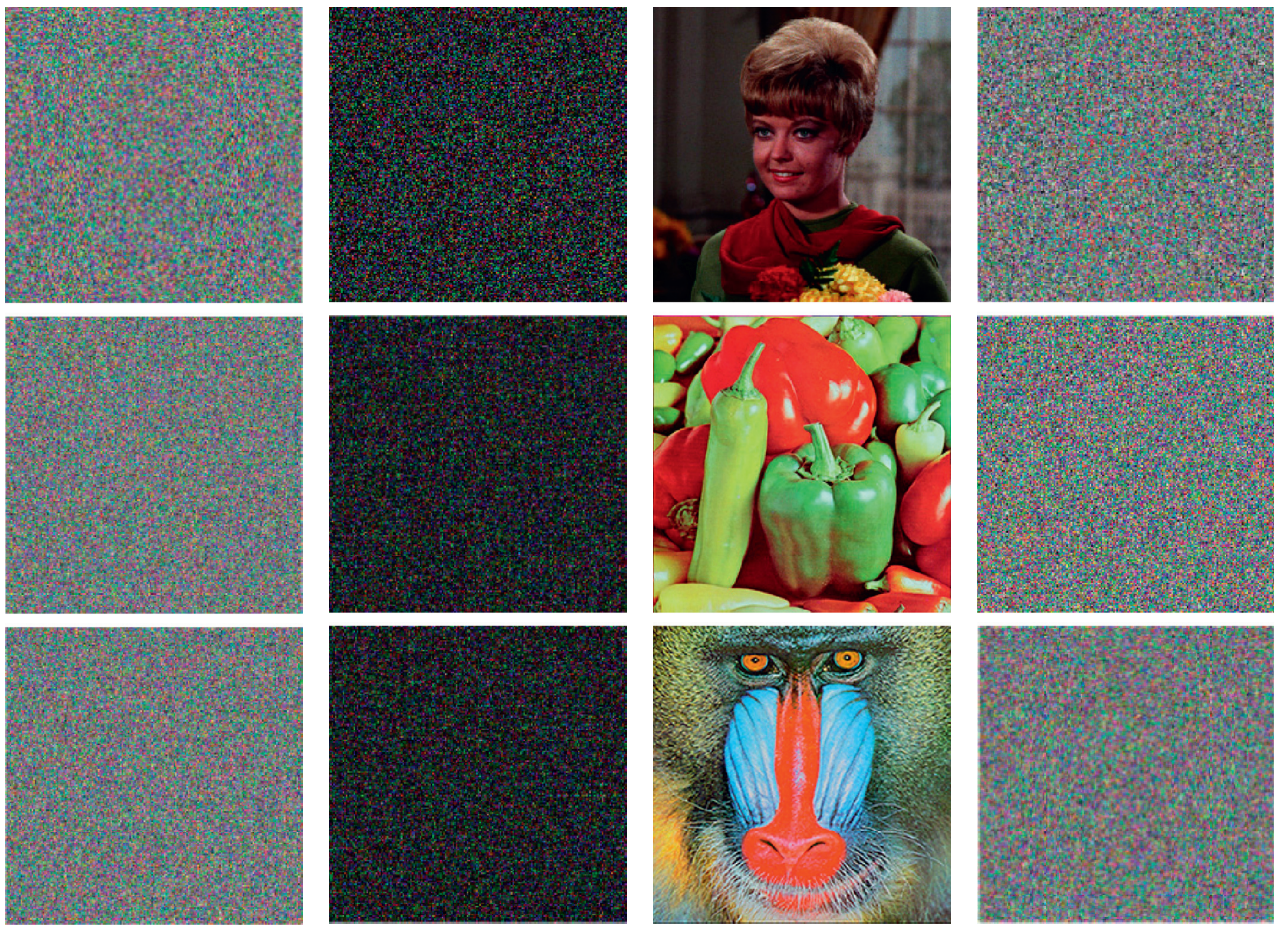
Table 7: Plaintext sensitivity test for different color images.

Algorithms	Images	Channels	NPCR (%)	UACI (%)	BACI (%)
The proposed scheme	Female	Red	99.6109	33.5025	26.8215
		Green	99.6039	33.4360	26.7813
		Blue	99.6149	33.4430	26.7754
	Peppers	Red	99.6005	33.4494	26.7600
		Green	99.6278	33.4537	26.7353
		Blue	99.6188	33.4529	26.7651
	Mandrill	Red	99.5941	33.4530	26.7913
		Green	99.6199	33.4472	26.7685
		Blue	99.6091	33.4596	26.7405
	Lena (256 × 256)	Red	99.6109	33.4783	26.7618
		Green	99.6213	33.4503	26.7713
		Blue	99.6004	33.4640	26.7794
	Tree (256 × 256)	Red	99.6062	33.4911	26.7929
		Green	99.5996	33.4628	26.7709
		Blue	99.6191	33.4522	26.7487
	Couple (256 × 256)	Red	99.6149	33.4617	26.7908
		Green	99.6083	33.4563	26.7618
		Blue	99.6114	33.4533	26.7512
Ref. [29]	Lena (256 × 256)	Red	99.5800	33.2700	—
		Green	99.5600	33.3600	—
		Blue	99.6400	33.5000	—
	Mandrill (256 × 256)	Red	99.6200	33.4700	—
		Green	99.6000	33.4800	—
		Blue	99.6000	33.4500	—
	Peppers (512 × 512)	Red	99.6000	33.3900	—
		Green	99.6000	33.4600	—
		Blue	99.6100	33.4000	—
Ref. [30]	Lena (256 × 256)	Red	99.6317	33.6783	—
		Green	99.6205	33.7999	—
		Blue	99.6211	33.6200	—
	Mandrill (256 × 256)	Red	99.6199	33.6484	—
		Green	99.6250	33.5908	—
		Blue	99.6273	33.6749	—
	Peppers (256 × 256)	Red	99.6202	33.6602	—
		Green	99.6192	33.6575	—
		Blue	99.6224	33.7314	—

Table 8: Comparison of the key space between our scheme and other chaos-based encryption algorithms.

Encryption algorithms	Key space
The proposed scheme	$(10^{15})^6 = 10^{90} \approx 2^{299}$
Ref. [8]	2^{106}
Ref. [15]	$10^{84} \approx 2^{279}$
Ref. [18]	2^{138}
Ref. [33]	2^{298}
Ref. [34]	$> 2^{208}$

In addition, Table 9 lists the average NPCR and UACI results for the three channels of the key sensitivity test for the plain images in Figure 5(a). It can be found that a slight change in each key will result in the change rate of more than 99.5% for the ciphered images. Their corresponding average NPCR and UACI results are 99.6124% and 33.4714%, which are closer to ideal than the values (NPCR = 99.6389% and UACI = 33.4189) in [35]. Thus, the proposed scheme has high key sensitivity.



(a) (b) (c) (d)

Figure 6: Key sensitivity test: (a) encrypted images with the key set K_1; (b) the pixel-by-pixel difference between Figures 5(b) and 6(a); (c) decrypted images from Figure 5(b) with the correct key set K_0; (d) decrypted images from Figure 5(b) with the wrong key set K_1.

Table 9: NPCR and UACI results for key sensitivity test.

Modified keys	Average NPCR (%)			Average UACI (%)		
	Female	Peppers	Mandrill	Female	Peppers	Mandrill
$X_{10} + 10^{-15}$	99.6132	99.6228	99.6164	33.4596	33.4677	33.4828
$X_{20} + 10^{-15}$	99.6088	99.6098	99.6062	33.4641	33.4606	33.4656
$X_{30} + 10^{-15}$	99.6112	99.6143	99.6124	33.4613	33.4860	33.4734
$r_1 + 10^{-15}$	99.6191	99.6115	99.6072	33.4762	33.4866	33.4674
$r_2 + 10^{-15}$	99.6124	99.6104	99.6120	33.4805	33.4678	33.4741
$r_3 + 10^{-15}$	99.6109	99.6156	99.6089	33.4691	33.4764	33.4665

5.4. Histogram Analysis. The image histogram is an important feature statistic of the image. A great encryption algorithm theoretically makes the histogram of the encrypted image evenly distributed. This prevents the adversary from getting any useful information. Figure 7 exhibits the histograms of the plain and ciphered Female, Peppers, and Mandrill images. It can be seen that the histograms of the ciphered images are fairly flat, and they are completely different from the histograms of the plain images.

Besides, the analysis of variance is a quantitative measure for the properties of histogram. The smaller the calculated variance value, the higher the uniformity of the image [36]. The specific calculation formula is as follows:

$$\text{variance}(Y) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\left(y_i - y_j\right)^2}{2}, \tag{25}$$

where y_i and y_j indicate the number of pixels corresponding to the gray values i and j, Y represents the vector set of the histogram and $Y = \{y_1, y_2, \ldots, y_{256}\}$, and n denotes the total number of gray values. Table 10 exhibits the variances of the histograms of the color plain and ciphered images in Figure 5. It can be discovered that the histogram variance values of the encrypted images are much smaller than those of the original images. Moreover, Table 11 displays the comparison of the mean histogram variance values of the

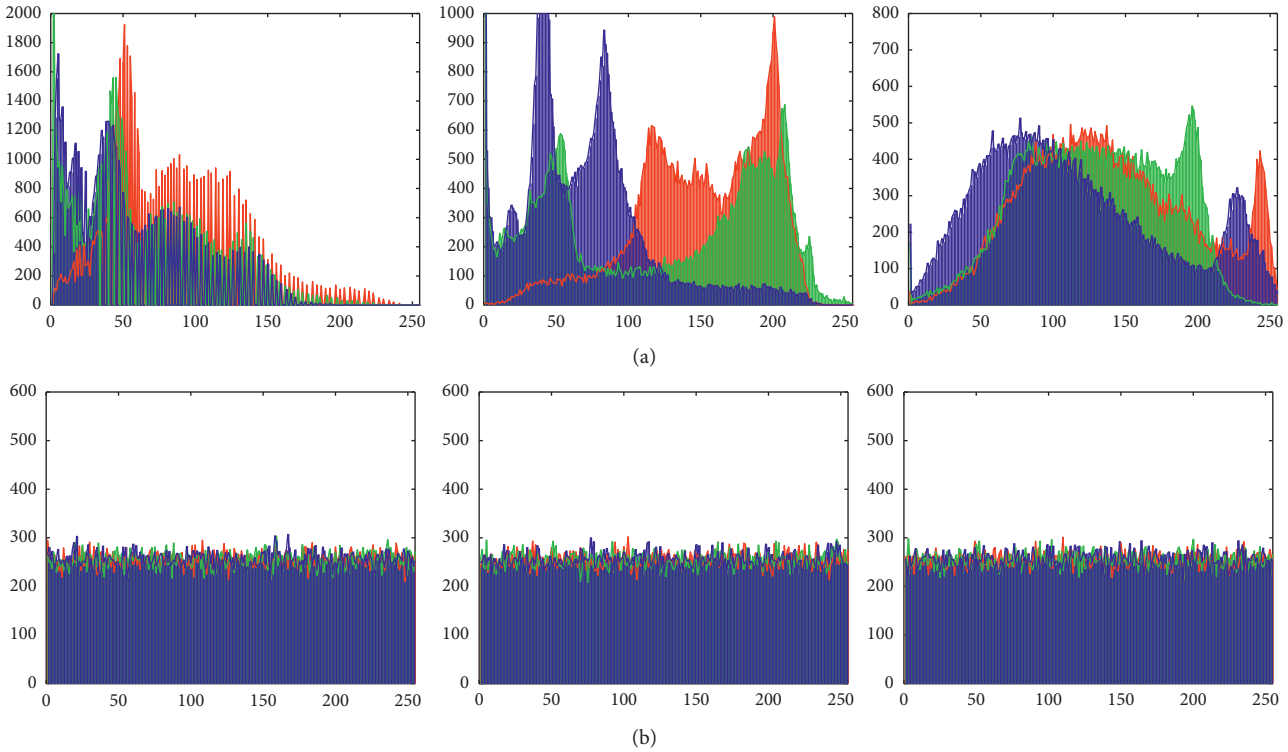


Figure 7: Histograms of the (a) plain images in Figure 5(a); (b) ciphered images in Figure 5(b).

three channels for the Lena image and the calculated variance results in [26, 36–38]. Obviously, the histogram variance values of our scheme are smaller, that is, the histograms of the ciphered images have better uniformity.

Then, we utilize the chi-square test to validate the uniformity for the pixel values distribution of the ciphered images [36]. Similar to the variances of the histograms, the smaller the result of the chi-square test, the better the uniformity of the ciphered image. Its definition is as follows:

$$\chi^2 = \sum_{i=0}^{255} \frac{(f_i - p_i)^2}{p_i}, \tag{26}$$

where f_i indicates the actual frequency of the pixel value i and p_i represents the predicted frequency of the pixel value i. The predicted frequency p_i can be calculated by

$$p_i = \frac{m \times n}{256}, \tag{27}$$

where $m \times n$ is the size of the image. Theoretically, the chi-square statistic with a significant level of 0.01 is 310.4574, while the chi-square statistic with a significant level of 0.05 is 293.2478. Obviously, the values of the chi-square test for six ciphered images in Table 12 are all smaller than the previous two theoretical values. Thus, it further illustrates the effectiveness of our scheme.

5.5. Correlation of Two Adjacent Pixels. As for plain images, there are strong correlations between their adjacent pixels. Thus, the purpose of encryption is to weaken this property. In the experiment, we randomly select 4000 pairs of adjacent pixels from the original images and corresponding encrypted images. Then, compute the correlation coefficients in horizontal, vertical, and diagonal directions as follows:

$$\begin{cases} E(x) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} x_i, \\ D(x) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} [x_i - E(x)]^2, \\ \mathrm{COV}(x, y) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} [x_i - E(x)][y_i - E(y)], \\ \rho_{xy} = \dfrac{\mathrm{COV}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}, \end{cases} \tag{28}$$

where $E(x)$,$D(x)$, and $\mathrm{COV}(x, y)$ indicate the mean value, variance, and covariance, respectively. Figure 8 plots the correlations between adjacent pixels of the plain Peppers image, and Figure 9 shows the correlation coefficients of the corresponding ciphered image. It can be seen from the comparison that the encryption makes the correlation between the pixels of the image significantly whittled. Moreover, Table 13 provides the correlation coefficients for the adjacent pixels of six color images with the size of 256×256 in three directions. It can be discovered that the results of the

Table 10: Histogram variance analysis of plain and ciphered images.

Images	Plain images			Ciphered images		
	Red	Green	Blue	Red	Green	Blue
Female	168960	157950	158040	233.4375	263.3125	267.3516
Peppers	54484	64353	106880	244.1172	258.2734	223.9297
Mandrill	20770	30655	16857	220.3125	240.2109	270.2656

Table 11: Histogram variance analysis of original Lena image and encrypted Lena image.

Images	Algorithms	Variance
Original Lena image (512×512)		6390323
Encrypted Lena image	The proposed scheme	953.7656
	Ref. [26]	977.02
	Ref. [36]	974.8
	Ref. [37]	1077.3
	Ref. [38]	1209.4

Table 12: Chi-square test of some ciphered images.

Images	Components			Results
	Red	Green	Blue	
Female	233.4375	253.3125	247.3516	Pass
Peppers	244.1172	238.2734	223.9297	Pass
Mandrill	220.3125	240.2109	240.2656	Pass
Couple (256×256)	234.2500	246.0313	233.3906	Pass
Tree (256×256)	222.5781	246.3047	235.5313	Pass
Lena (512×512)	245.8965	230.9863	252.1680	Pass
Airplane (512×512)	240.7422	257.8555	235.8145	Pass

ciphered images are close to 0. Furthermore, Table 14 reveals that our encryption scheme has superior performance compared to some recent encryption algorithms.

5.6. Information Entropy Analysis. Information entropy is expressed as the probability of discrete random events, which is used to measure the randomness of a system. It can be defined as

$$H(s) = -\sum_{i=0}^{2^n-1} P(s_i)\log_2[P(s_i)], \tag{29}$$

where P_{s_i} denotes the probability that the gray value s_i may appear. For a grayscale image with data field of $[0, 255]$, the maximum value of information entropy is 8. Therefore, once the calculated value is pretty close to 8, it proves that the proposed algorithm is quite safe. We have computed the information entropy of six different color images with the size of 256×256 and their ciphered versions in Table 15. From the table, the information entropy results of ciphered images are approaching the ideal value 8. In addition, Table 16 exhibits the information entropy of the original Lena image and encrypted Lena image by utilizing our scheme and some other encryption algorithms. It can be clearly seen that our cryptosystem is closer to the desired state and has better randomness.

5.7. Noise Attack Analysis. In practical applications, noise interference is inevitable. An outstanding encryption algorithm has the ability to resist noise attack. In our experiment, the encrypted Mandrill image in Figure 5(b) is contaminated by Salt & Pepper noise, Gaussian noise, and speckle noise with different densities, respectively. The simulation results are exhibited in Figures 10–12. It can be observed that although the noise is increasing, the decrypted images can still be discerned. Moreover, we also tested the average NPCR and UACI values for the three channels of the original Mandrill image and deciphered Mandrill image under these different noises. The results are listed in Tables 17–19. It can be found that all the NPCR values are less than 99%, and all the UACI values are less than 20%. Hence, the proposed method is strongly robust against disparate noises.

5.8. Cropping Attack Analysis. Encrypted images are subject to cropping attack during transmission and may be partially damaged. Nevertheless, digital images allow a certain extent of distortion on the transmission channel. As long as the information to be conveyed by the image can be discriminated visually, it proves that the encryption algorithm has excellent anticropping attack capability. Figure 13 displays ciphered House image of size 256×256 with data cuts in different sizes, and their corresponding deciphered images. Still, most of the pictorial information is available from the

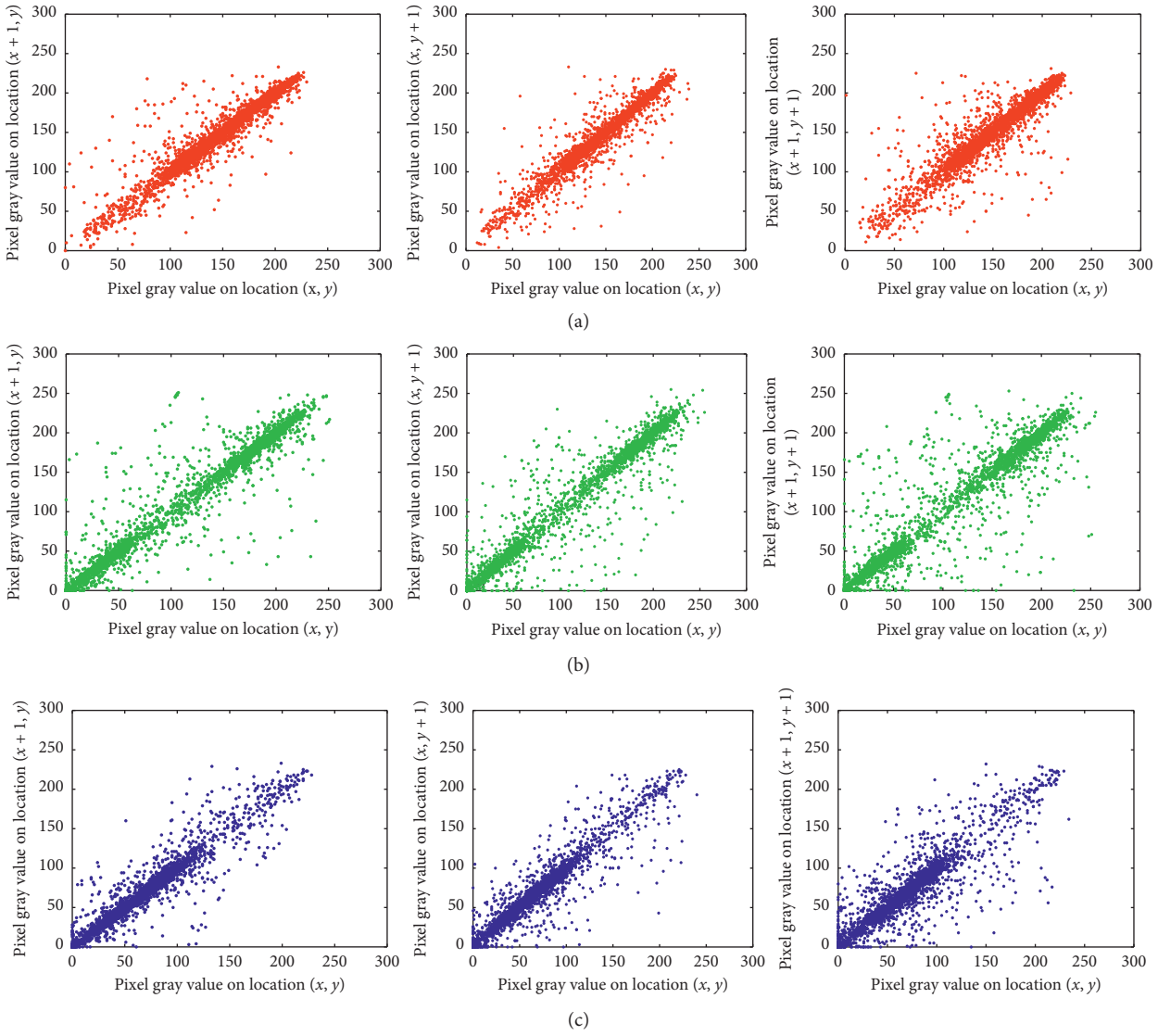


Figure 8: Correlation coefficients of plain Peppers image in all directions: (a) R channel; (b) G channel; (c) B channel.

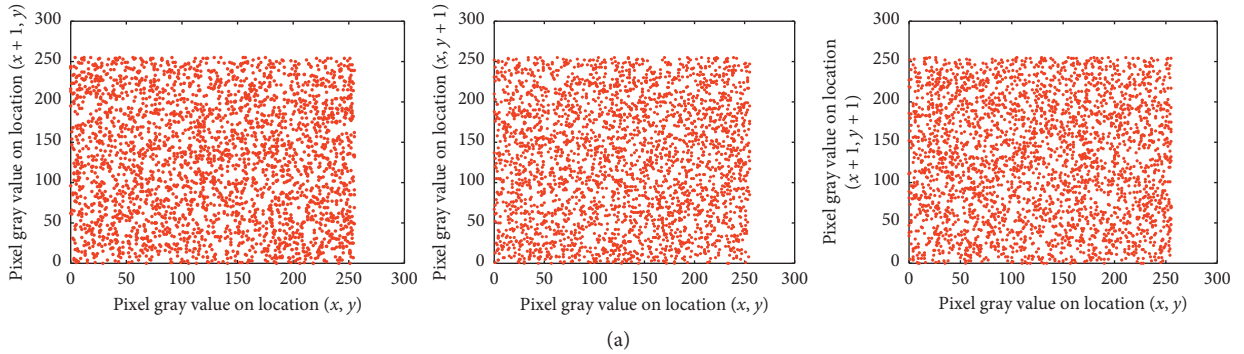


Figure 9: Continued.

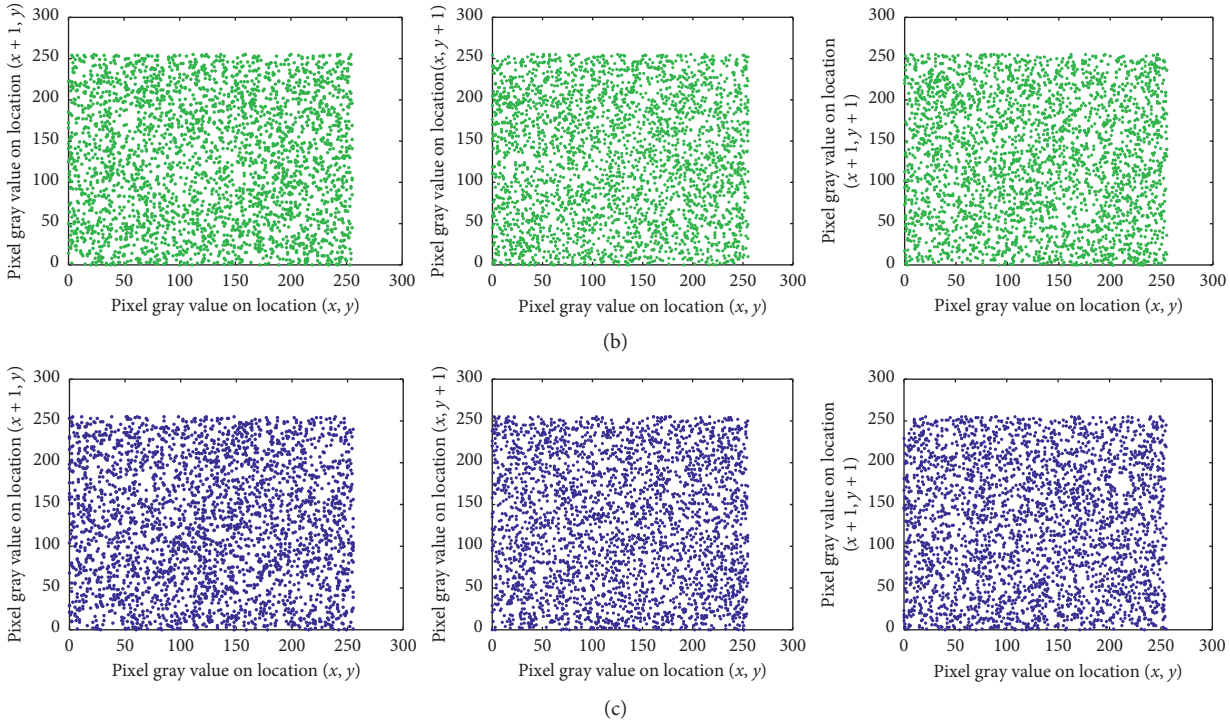


Figure 9: Correlation coefficients of ciphered Peppers image in all directions: (a) R channel; (b) G channel; (c) B channel.

Table 13: Average correlation coefficients of three channels for plain and ciphered images.

Images	Plain images			Ciphered images		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Female	0.9747	0.9621	0.9511	0.0078	0.0021	0.0004
Peppers	0.9647	0.9701	0.9602	0.0004	−0.0035	−0.0021
Mandrill	0.9188	0.8530	0.8534	−0.0031	0.0048	0.0005
Tree	0.9406	0.9551	0.9157	−0.0012	−0.0046	0.0006
Couple	0.9131	0.9011	0.9375	0.0074	0.0035	0.0013
House	0.9328	0.9676	0.9114	−0.0027	0.0047	0.0012

Table 14: Correlation coefficients of original Lena image and encrypted Lena image.

Images	Algorithms	Directions		
		Horizontal	Vertical	Diagonal
Original Lena image (256 × 256)		0.9731	0.9881	0.9661
Encrypted Lena image	The proposed scheme	−0.0034	0.0021	−0.0003
	Ref. [14]	0.0056	0.0065	−0.0073
	Ref. [19]	0.0036	0.0023	0.0039
	Ref. [33]	−0.0068	−0.0054	0.0010
	Ref. [39]	0.0040	0.0011	0.0008
	Ref. [40]	0.0059	−0.0042	0.0180
	Ref. [41]	−0.0168	0.0445	−0.0022
	Ref. [42]	−0.0003	−0.0013	−0.0066

decrypted images. This shows that the proposed scheme can effectively resist cropping attack.

5.9. Known Plaintext Attack and Chosen Plaintext Attack Analysis. Since the cryptosystems whose key streams are unrelated to the plain images are vulnerable to chosen plaintext attack and known plaintext attack [21], we design the initial keys of our scheme to be updated by the bit-planes recombination of the plain images. Under the circumstances, we can guarantee that different images are encrypted by diverse key streams, and the attackers cannot obtain

Table 15: Information entropy analysis of plain and ciphered images.

Images	Plain images			Ciphered images		
	Red	Green	Blue	Red	Green	Blue
Female	6.4200	6.4457	6.3807	7.9974	7.9971	7.9971
Peppers	7.3449	7.5607	7.1003	7.9973	7.9972	7.9975
Mandrill	7.7255	7.5618	7.8031	7.9976	7.9974	7.9970
Tree	7.2104	7.4136	6.9207	7.9976	7.9971	7.9970
Couple	6.2499	5.9642	5.9309	7.9972	7.9973	7.9972
House	6.4311	6.5389	6.2320	7.9975	7.9968	7.9973

Table 16: Comparison of the information entropy of the original Lena image and encrypted Lena image.

Information entropy	Red	Green	Blue
Original Lena image (256×256)	7.3140	7.6394	7.0506
The proposed algorithm	7.9971	7.9972	7.9972
Ref. [39]	7.9278	7.9744	7.9705
Ref. [41]	7.9897	7.9877	7.9896
Ref. [43]	7.9988	7.9967	7.9990
Ref. [44]	7.9791	7.9802	7.9827

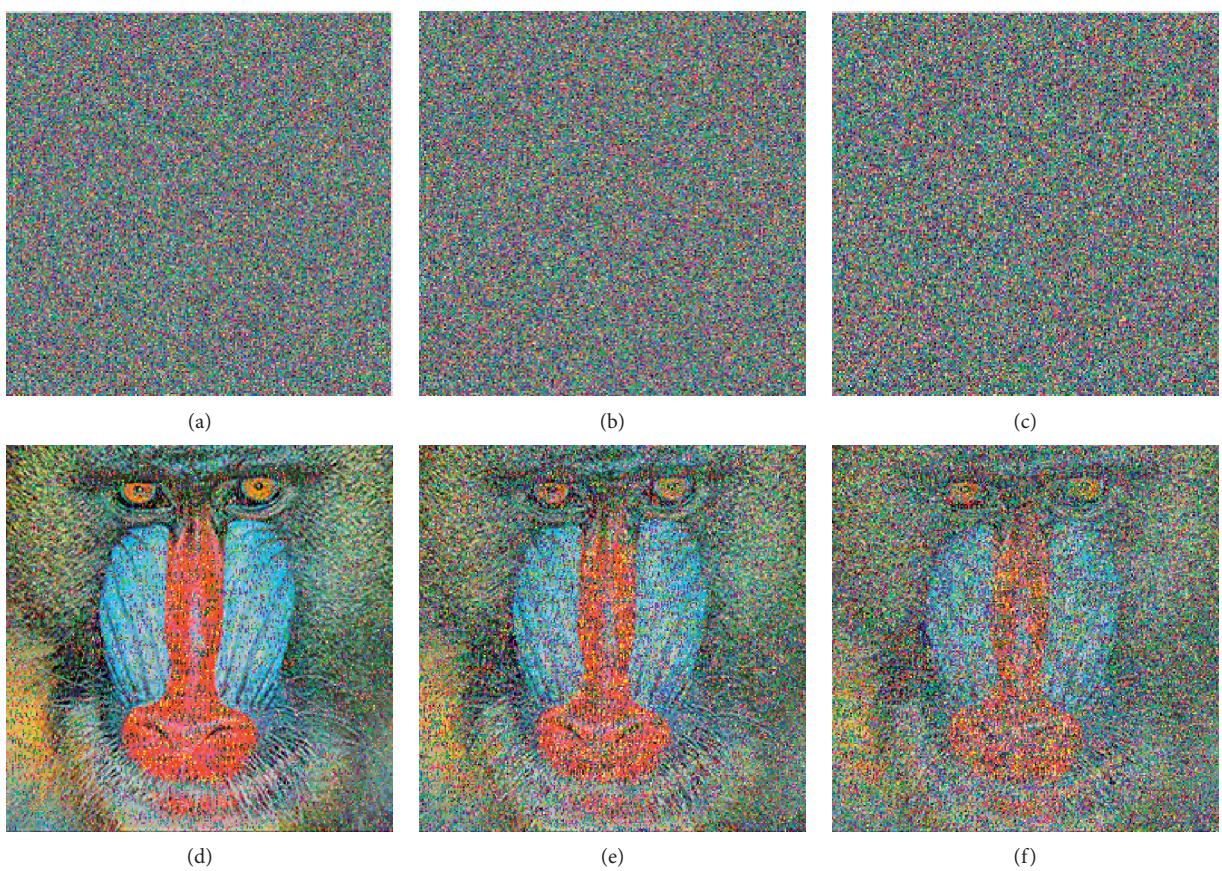
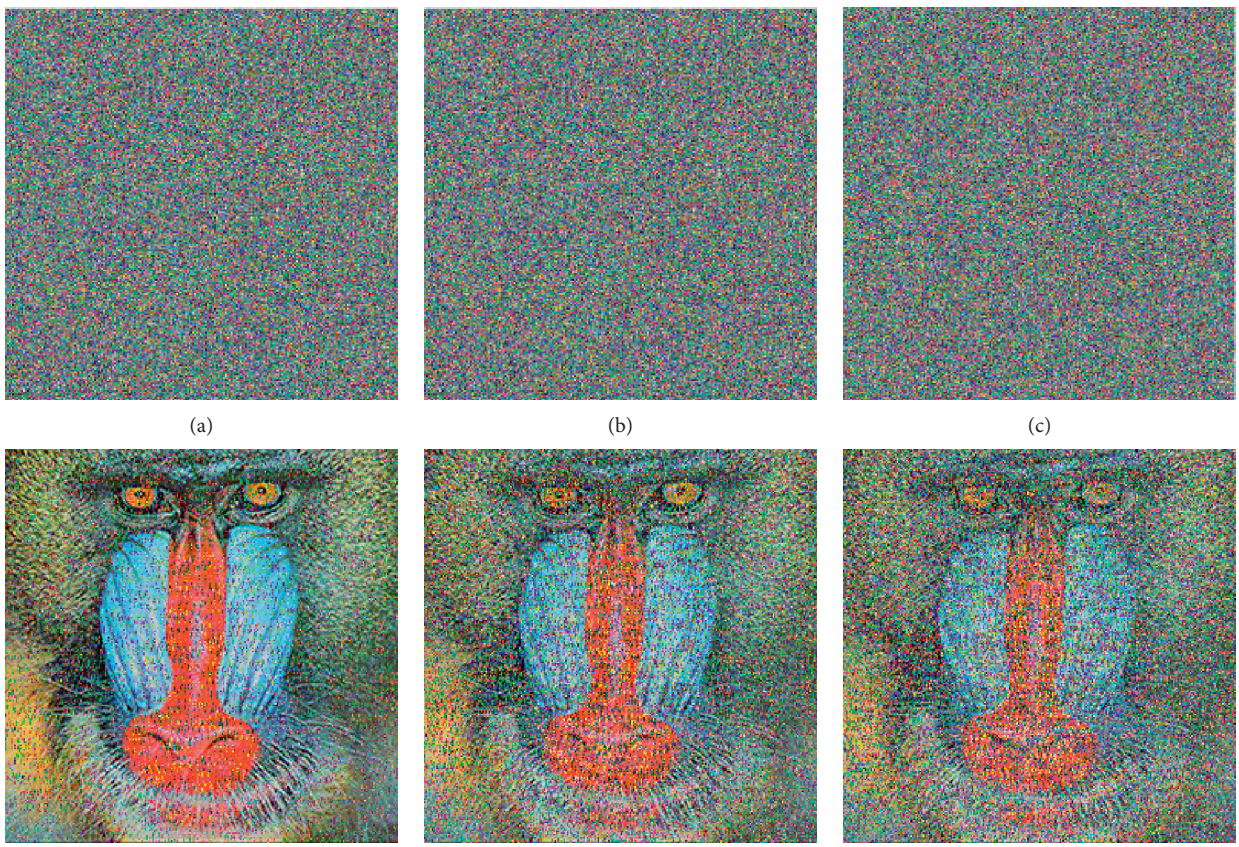


Figure 10: Experiment of Salt & Pepper noise attacks: (a) encrypted Mandrill image attacked by Pepper & Salt noise with density 0.05; (b) encrypted Mandrill image attacked by Pepper & Salt noise with density 0.1; (c) encrypted Mandrill image attacked by Pepper & Salt noise with density 0.2; (d) decrypted Mandrill image of a; (e) decrypted Mandrill image of b; (f) decrypted Mandrill image of c.

serviceable information by selecting certain special images. Thus, our scheme can effectively resist the known plaintext attack and chosen plaintext attack.

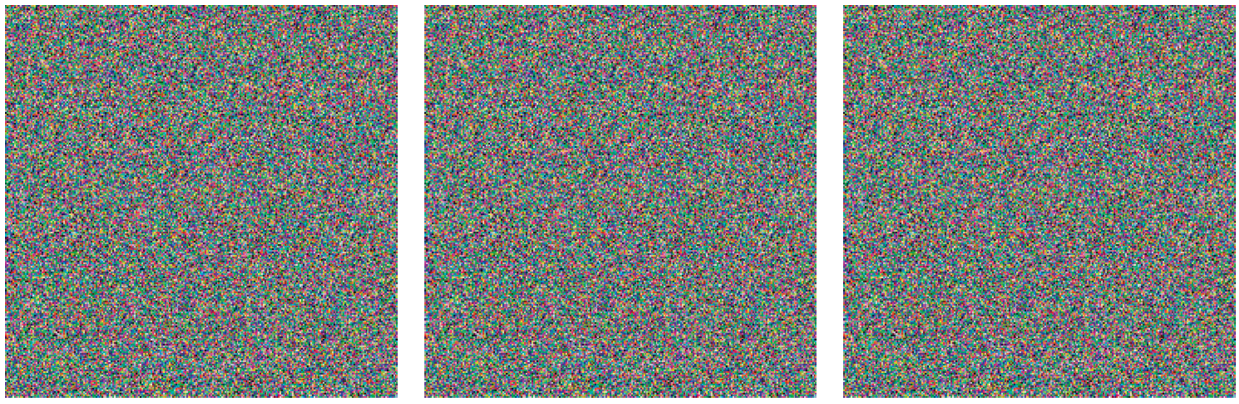
5.10. MSE and PSNR Analysis. Mean square error (MSE) is a relatively straightforward method to measure the average error. When evaluating an encryption algorithm, verification



(a) (b) (c)

(d) (e) (f)

Figure 11: Experiment of Gaussian noise attacks: (a) encrypted Mandrill image attacked by Gaussian noise with mean value 0 and variance value 0.0001; (b) encrypted Mandrill image attacked by Gaussian noise with mean value 0 and variance value 0.0003; (c) encrypted Mandrill image attacked by Gaussian noise with mean value 0 and variance value 0.0005; (d) decrypted Mandrill image of a; (e) decrypted Mandrill image of b; (f) decrypted Mandrill image of c.



(a) (b) (c)

Figure 12: Continued.

(d) (e) (f)

Figure 12: Experiment of speckle noise attacks: (a) encrypted Mandrill image attacked by speckle noise with density 0.00005; (b) encrypted Mandrill image attacked by speckle noise with density 0.0001; (c) encrypted Mandrill image attacked by speckle noise with density 0.0005; (d) decrypted Mandrill image of a; (e) decrypted Mandrill image of b; (f) decrypted Mandrill image of c.

Table 17: NPCR and UACI values for the original Mandrill image and deciphered Mandrill image under Salt & Pepper noise.

Images	Density	Average NPCR	Average UACI
Figures 5(a) and 10(d)	0.05	26.0681	7.7270
Figures 5(a) and 10(e)	0.1	46.2418	13.6364
Figures 5(a) and 10(f)	0.2	62.4237	18.5213

Table 18: NPCR and UACI values for the original Mandrill image and deciphered Mandrill image under Gaussian noise.

Images	Mean	Variance	Average NPCR	Average UACI
Figures 5(a) and 11(d)	0	0.0001	98.4116	8.8154
Figures 5(a) and 11(e)	0	0.0003	98.7732	11.7899
Figures 5(a) and 11(f)	0	0.0005	98.9624	12.7496

Table 19: NPCR and UACI values for the original Mandrill image and deciphered Mandrill image under speckle noise.

Images	Density	Average NPCR	Average UACI
Figures 5(a) and 12(d)	0.00005	94.5068	8.5631
Figures 5(a) and 12(e)	0.0001	95.5780	11.3172
Figures 5(a) and 12(f)	0.0005	97.4060	19.3016

is performed by comparing the MSE results between plain images and ciphered images, while also comparing the MSE results between plain images and deciphered images [37]. In addition, the larger the MSE value between plain images and their ciphers, the better the encryption effect. The definitions of MSE are as follows:

$$\mathrm{MSE}_{PC} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (P(i,j) - C(i,j))^2,$$

$$\mathrm{MSE}_{PD} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (P(i,j) - D(i,j))^2, \quad (30)$$

where $P(i,j)$ denotes plain images, $C(i,j)$ represents ciphered images, and $D(i,j)$ means deciphered images.

Peak signal-to-noise ratio (PSNR) is utilized to appraise the distortion of images. In contrast to MSE, the smaller the PSNR value between plain images and their ciphers, the greater the difference. The definitions of PSNR are as follows:

$$\mathrm{PSNR}_{PC} = 20 \log_{10} \frac{L}{\sqrt{\mathrm{MSE}_{PC}}},$$

$$\mathrm{PSNR}_{PD} = 20 \log_{10} \frac{L}{\sqrt{\mathrm{MSE}_{PD}}}, \quad (31)$$

where L is the maximum gray value. Table 20 shows the MSE and PSNR results between some plain images and their ciphered images. Obviously, the MSE results are quite large, and the PSNR values are pretty small. Moreover, the average MSE and PSNR values for the three channels of the Lena image with the size of 512×512 are 10116.3 and 8.0935,

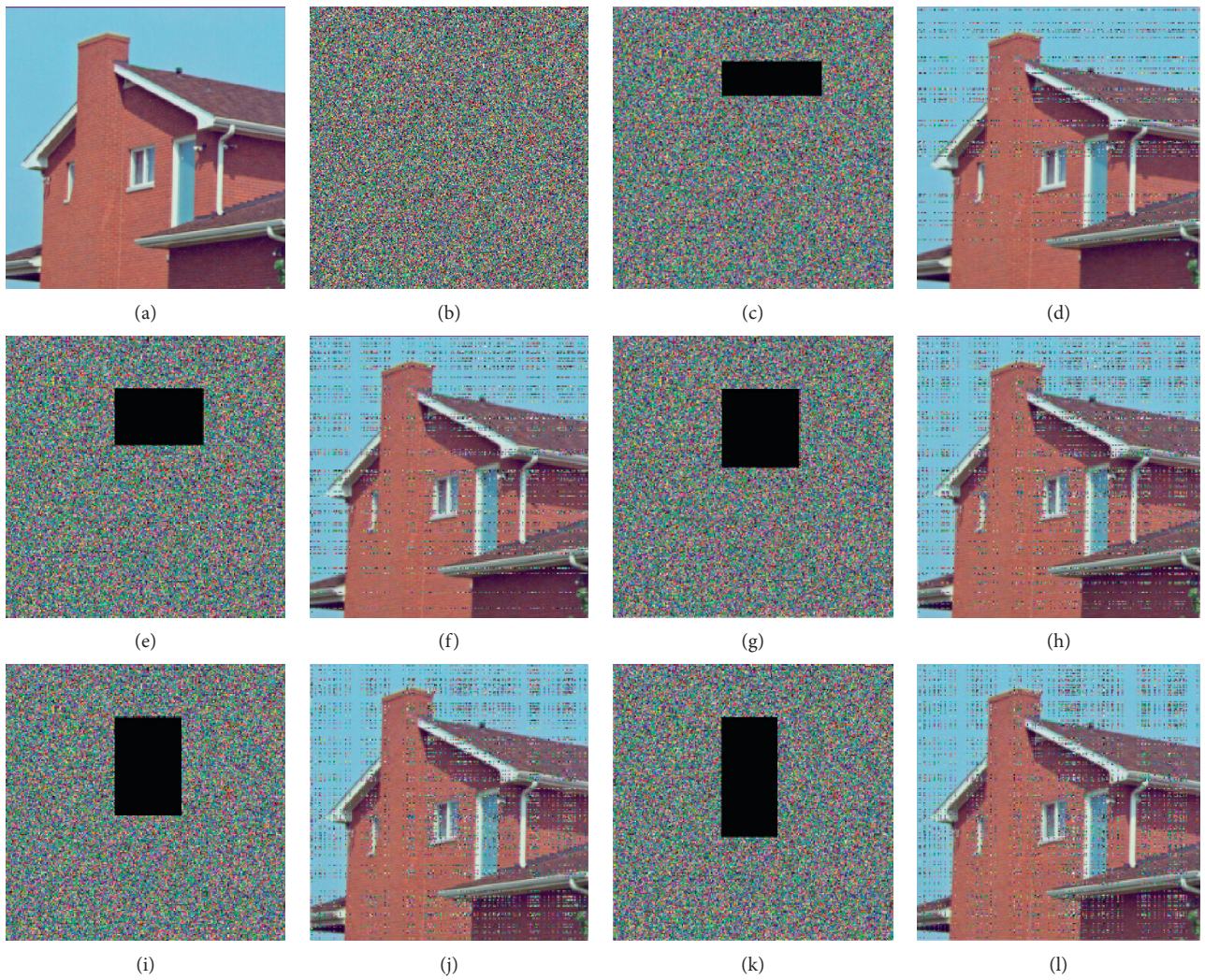


Figure 13: Test of image cropping attacks: (a) original House image; (b) encrypted House image; (c) encrypted House image with a 30×90 data cut; (d) decrypted House image of (c); (e) encrypted House image with a 50×80 data cut; (f) decrypted House image of (e); (g) encrypted House image with a 70×70 data cut; (h) decrypted House image of (g); (i) encrypted House image with a 90×60 data cut; (j) decrypted House image of (i); (k) encrypted House image with a 110×50 data cut; (l) decrypted House image of (k).

Table 20: MSE and PSNR values between plain and ciphered images.

Images	MSE			PSNR		
	Red	Green	Blue	Red	Green	Blue
Female (256×256)	10081	12935	13536	8.0959	7.0131	6.8160
Peppers (256×256)	8087.8	11309	11211	9.0525	7.5967	7.6344
Mandrill (256×256)	8698.6	7871.9	9603.6	8.7363	9.1700	8.3065
Lena (256×256)	10791	9139.3	1067.2	7.8004	8.5217	7.8483
Lena (512×512)	10683	9035.8	10630	7.8440	8.5711	7.8655
Tree (256×256)	8801.6	11387	9718.1	8.6852	7.5667	8.2550
Couple (256×256)	14005	16095	16246	6.6680	6.0638	6.0233

respectively. They are better than the test results (MSE = 9081.2 and PSNR = 8.2203) of the Lena image in [37]. In theory, the decrypted image is identical to the original image, thus the MSE value is 0 and the PSNR value is infinite. It can be found from Table 21 that the values of the test are consistent with the theoretical situation.

5.11. Time Complexity Analysis. In order to analyze the time cost, we elaborate on the computational complexity. In the proposed scheme, the computational cost is relevant to the encryption steps. First, the iteration of the SLM, CLM, and SCM will produce the time complexity of $O(3 \times m \times n)$. Then, the time complexity of pixel-level scrambling is also

Table 21: MSE and PSNR values between plain and deciphered images.

Images	MSE			PSNR		
	Red	Green	Blue	Red	Green	Blue
Female (256 × 256)	0	0	0	∞	∞	∞
Peppers (256 × 256)	0	0	0	∞	∞	∞
Mandrill (256 × 256)	0	0	0	∞	∞	∞
Lena (256 × 256)	0	0	0	∞	∞	∞
Lena (512 × 512)	0	0	0	∞	∞	∞
Tree (256 × 256)	0	0	0	∞	∞	∞
Couple (256 × 256)	0	0	0	∞	∞	∞

Table 22: Comparison of the time complexity of different algorithms.

Algorithms	Time complexity
The proposed scheme	6 mn
Ref. [36]	4 mn
Ref. [45]	8 mn
Ref. [46]	24 mn

$O(3 \times m \times n)$. Subsequently, the time complexity of the sequence conversions is $O(6 \times m \times n)$. After DNA coding, the time complexity of the DNA calculation determined by the sequence generated by the SLM is $O(3 \times m \times n)$. Analogously, the time complexity of decoding DNA matrixes is $O(3 \times m \times n)$. Ultimately, the time complexity of multiplication over the Galois field is $O(3 \times m \times n)$. Therefore, the total time complexity of our scheme is $O(6 \times m \times n)$. Furthermore, it can be seen from the time complexity results in Table 22 that although our scheme has higher time complexity than the encryption algorithms in [36], its computational complexity is lower than that of the algorithms in [45] and [46]. Thus, it can be proved that our scheme is effective.

6. Conclusions

In this paper, we introduce a novel chaotic color image encryption scheme based on DNA coding calculations and arithmetic over the Galois field. Firstly, three 1D chaotic maps with better chaotic properties are obtained by improving the classical 1D chaotic maps, and we use them as the secret keys for the cryptosystem. Meanwhile, the application of plain images to update the initial values protects our scheme from the threat of chosen plaintext attack and known plaintext attack. In order to increase the degree of diffusion, the scheme also adds the coding calculations of DNA sequences and multiplication over the Galois fields GF(17). At last, the simulation results verify that the proposed algorithm has excellent performance. The future work is to research the fast encryption scheme based on chaos, which can be applied in real-time communication scenarios such as telemedicine.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (nos. 61203004 and 61306142), Natural Science Foundation of Heilongjiang Province (no. F201220), and Fundamental Research Funds for the Central Universities (no. 3072019CFG0802).

References

[1] J. S. A. E. Fouda, J. Y. Effa, S. L. Sabat, and M. Ali, "A fast chaotic block cipher for image encryption," *Communications in Nonlinear Science & Numerical Simulation*, vol. 19, no. 3, pp. 578–588, 2014.

[2] D. Ravichandran, P. Praveenkumar, J. B. Balaguru Rayappan, and R. Amirtharajan, "Chaos based crossover and mutation for securing DICOM image," *Computers in Biology and Medicine*, vol. 72, pp. 170–184, 2016.

[3] P. G. Pashakolaee, H. S. Shahhoseini, and M. Mollajafari, "Hyper-chaotic Feeded GA (HFGA): a reversible optimization technique for robust and sensitive image encryption," *Multimedia Tools and Applications*, vol. 77, no. 16, pp. 20385–20414, 2018.

[4] B. Norouzi, S. Mirzakuchaki, S. M. Seyedzadeh, and M. R. Mosavi, "A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process," *Multimedia Tools and Applications*, vol. 71, no. 3, pp. 1469–1497, 2014.

[5] J. Fridrich, "Image encryption based on chaotic maps," in *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 2, pp. 1105–1110, Orlando, FL, USA, October 1997.

[6] X. Chen and C.-J. Hu, "Adaptive medical image encryption algorithm based on multiple chaotic mapping," *Saudi Journal of Biological Sciences*, vol. 24, no. 8, pp. 1821–1827, 2017.

[7] M. F. Haroun and T. A. Gulliver, "Real-time image encryption using a low-complexity discrete 3D dual chaotic cipher," *Nonlinear Dynamics*, vol. 82, no. 3, pp. 1–13, 2015.

[8] C. Li, G. Luo, K. Qin, and C. Li, "An image encryption scheme based on chaotic tent map," *Nonlinear Dynamics*, vol. 87, no. 1, pp. 127–133, 2017.

[9] K. A. K. Patro and B. Acharya, "A simple, secure, and time-efficient bit-plane operated bit-level image encryption scheme

using 1-D chaotic maps," *Innovations in Soft Computing and Information Technology*, vol. 3, pp. 261–278, 2019.

[10] S. Anwar and S. Meghana, "A pixel permutation based image encryption technique using chaotic map," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27569–27590, 2019.

[11] Z. Bashir, T. Rashid, and S. Zafar, "Hyperchaotic dynamical system based image encryption scheme with time-varying delays," *Pacific Science Review A: Natural Science and Engineering*, vol. 18, no. 3, pp. 254–260, 2016.

[12] M. L. Sahari and I. Boukemara, "A pseudo-random numbers generator based on a novel 3D chaotic map with an application to color image encryption," *Nonlinear Dynamics*, vol. 94, no. 1, pp. 723–744, 2018.

[13] X.-J. Tong, M. Zhang, Z. Wang, and J. Ma, "A joint color image encryption and compression scheme based on hyper-chaotic system," *Nonlinear Dynamics*, vol. 84, no. 4, pp. 2333–2356, 2016.

[14] X. Wang and H. L. Zhang, "A novel image encryption algorithm based on genetic recombination and hyper-chaotic systems," *Nonlinear Dynamics*, vol. 83, no. 1-2, pp. 333–346, 2016.

[15] Y. Zhou, L. Bao, and C. L. P. Chen, "A new 1D chaotic system for image encryption," *Signal Processing*, vol. 97, no. 7, pp. 172–182, 2014.

[16] S. Dhall, S. K. Pal, and K. Sharma, "Cryptanalysis of image encryption scheme based on a new 1D chaotic system," *Signal Processing*, vol. 146, pp. 22–32, 2018.

[17] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D Sine Logistic modulation map for image encryption," *Information Sciences*, vol. 297, pp. 80–94, 2015.

[18] C. Pak and L. Huang, "A new color image encryption using combination of the 1D chaotic map," *Signal Processing*, vol. 138, pp. 129–137, 2017.

[19] Z. Qiang, L. Guo, and X. Wei, "Image encryption using DNA addition combining with chaotic maps," *Mathematical & Computer Modelling*, vol. 52, no. 11-12, pp. 2028–2035, 2010.

[20] L. Li, Z. Qiang, X. Wei, and C. Zhou, "Image encryption algorithm based on chaotic modulation of Arnold dual scrambling and DNA computing," *Advanced Science Letters*, vol. 4, no. 11, pp. 3537–3542, 2011.

[21] Z. Yong, "Cryptanalysis of an image encryption algorithm based on chaotic modulation of Arnold dual scrambling and DNA computing," *Advanced Science Focus*, vol. 2, no. 1, pp. 67–82, 2014.

[22] P. Zhen, G. Zhao, L. Min, and X. Jin, "Chaos-based image encryption scheme combining DNA coding and entropy," *Multimedia Tools and Applications*, vol. 75, no. 11, pp. 6303–6319, 2016.

[23] S. Xin, W. Li, and H. Hu, "Cryptanalysis of a chaos-based image encryption scheme combining DNA coding and entropy," *Multimedia Tools and Applications*, vol. 76, no. 12, pp. 1–13, 2016.

[24] M. Babaei, "A novel text and image encryption method based on chaos theory and DNA computing," *Natural Computing*, vol. 12, no. 1, pp. 101–107, 2013.

[25] H. Liu, X. Wang, and A. Kadir, "Image encryption using DNA complementary rule and chaotic maps," *Applied Soft Computing*, vol. 12, no. 5, pp. 1457–1466, 2012.

[26] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using DNA sequence operations," *Optics and Lasers in Engineering*, vol. 88, pp. 197–213, 2017.

[27] X. Huang, "Image encryption algorithm using chaotic Chebyshev generator," *Nonlinear Dynamics*, vol. 67, no. 4, pp. 2411–2417, 2012.

[28] A. A. Abd El-Latif and X. Niu, "A hybrid chaotic system and cyclic elliptic curve for image encryption," *AEU—International Journal of Electronics and Communications*, vol. 67, no. 2, pp. 136–143, 2013.

[29] K. Majid and M. Fawad, "A novel chaotic image encryption technique based on multiple discrete dynamical maps," *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 26203–26222, 2019.

[30] A. Belazi, M. Khan, A. A. A. El-Latif, and S. Belghith, "Efficient cryptosystem approaches: S-boxes and permutation-substitution-based encryption," *Nonlinear Dynamics*, vol. 87, no. 1, pp. 337–361, 2017.

[31] Y. Zhang, "The unified image encryption algorithm based on chaos and cubic S-Box," *Information Sciences*, vol. 450, pp. 361–377, 2018.

[32] K. A. K. Patro and B. Acharya, "Secure multi-level permutation operation based multiple colour image encryption," *Journal of Information Security and Applications*, vol. 40, pp. 111–133, 2018.

[33] S. Sun, "A novel hyperchaotic image encryption scheme based on DNA encoding, pixel-level scrambling and bit-level scrambling," *IEEE Photonics Journal*, vol. 10, no. 2, pp. 1–14, 2018.

[34] A. Belazi, A. A. Abd El-Latif, A.-V. Diaconu, R. Rhouma, and S. Belghith, "Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms," *Optics and Lasers in Engineering*, vol. 88, pp. 37–50, 2017.

[35] A. A. Abd El-Latif, L. Li, N. Wang, Q. Han, and X. Niu, "A new approach to chaotic image encryption based on quantum chaotic system, exploiting color spaces," *Signal Processing*, vol. 93, no. 11, pp. 2986–3000, 2013.

[36] K. A. K. Patro, B. Acharya, and V. Nath, "Secure, lossless, and noise-resistive image encryption using chaos, hyper-chaos, and DNA sequence operation," *IETE Technical Review*, pp. 1–23, 2019.

[37] K. A. K. Patro, B. Acharya, and V. Nath, "Secure multilevel permutation-diffusion based image encryption using chaotic and hyper-chaotic maps," *Microsystem Technologies*, vol. 25, no. 12, pp. 4593–4607, 2019.

[38] X. Chai, K. Yang, and Z. Gan, "A new chaos-based image encryption algorithm with dynamic key selection mechanisms," *Multimedia Tools and Applications*, vol. 76, no. 7, pp. 9907–9927, 2017.

[39] A. Kadir, A. Hamdulla, and W.-Q. Guo, "Color image encryption using skew tent map and hyper chaotic system of 6th-order CNN," *Optik*, vol. 125, no. 5, pp. 1671–1675, 2014.

[40] R. Enayatifar, H. J. Sadaei, A. H. Abdullah, M. Lee, and I. F. Isnin, "A novel chaotic based image encryption using a hybrid model of deoxyribonucleic acid and cellular automata," *Optics and Lasers in Engineering*, vol. 71, pp. 33–41, 2015.

[41] L. Liu, Q. Zhang, and X. Wei, "A RGB image encryption algorithm based on DNA encoding and chaos map," *Computers & Electrical Engineering*, vol. 38, no. 5, pp. 1240–1248, 2012.

[42] B. Abd-El-Atty, A. A. Abd El-Latif, and S. Venegas-Andraca, "An encryption protocol for NEQR images based on one-particle quantum walks on a circle," *Quantum Information Processing*, vol. 18, no. 9, p. 272, 2019.

[43] A. Belazi, A. A. Abd El-Latif, and S. Belghith, "A novel image encryption scheme based on substitution-permutation network and chaos," *Signal Processing*, vol. 128, pp. 155–170, 2016.

[44] H. Liu and X. Wang, "Color image encryption using spatial bit-level permutation and high-dimension chaotic system,"

Optics Communications, vol. 284, no. 16-17, pp. 3895–3903, 2011.

[45] L. Xu, Z. Li, J. Li, and W. Hua, "A novel bit-level image encryption algorithm based on chaotic maps," *Optics and Lasers in Engineering*, vol. 78, pp. 17–25, 2016.

[46] X. Wu, H. Kan, and J. Kurths, "A new color image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps," *Applied Soft Computing*, vol. 37, pp. 24–39, 2015.