Hindawi
International Journal of Optics
Volume 2019, Article ID 3594534, 13 pages
https://doi.org/10.1155/2019/3594534

Research Article

Entropy-Based Block Scrambling Image Encryption Using DES Structure and Chaotic Systems

Xuncai Zhang, Lingfei Wang, Guangzhao Cui, and Ying Niu

School of Electrics and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

Correspondence should be addressed to Ying Niu; niuying@zzuli.edu.cn

Received 28 May 2019; Accepted 30 July 2019; Published 15 August 2019

Academic Editor: Chenggen Quan

Traditional encryption algorithms are inefficient when applied to image encryption because image data have the characteristics of large data sizes and strong correlations between adjacent pixels. The shortcomings of the traditional Data encryption standard (DES) encryption algorithm when applied to image encryption are analyzed, and a new image encryption algorithm based on the traditional DES encryption algorithm model, chaotic systems, DNA computing, and select cipher-text output is proposed. Select cipher-text output selects cipher image with the biggest entropy, and it can increase the randomness of cipher image and reduce the risk of encryption system being broken down. This algorithm overcomes the shortcomings of high computational complexity and inconvenient key management that the traditional text encryption algorithm has when applied to image encryption. The experimental results show that the security of this algorithm is verified by analyzing the information entropy, image correlation of adjacent pixels and other indexes. At the same time, this algorithm passes the noise attack test and the occlusion attack test, so it can resist common attacks.

1. Introduction

In modern society, information security issues affect almost all aspects of human production and life. The problem of information leakage is becoming increasingly serious. Developing mechanisms to effectively protect the security of information has become a hot research field. Traditional encryption algorithms such as RSA and DES have been widely used in the encryption of text information [1, 2], but with the advancement of the semiconductor industry and computer science, human computing capabilities have achieved rapid progress. Traditional encryption algorithms are facing the risk of being cracked. At the same time, although the computing capabilities of computers have been greatly improved, humans still need higher encryption speeds. The size of image data is much larger than that of text data, and the correlation between adjacent pixels is very strong. Therefore, there is a need to develop more efficient and secure image encryption algorithms.

There are two methods for image encryption: scrambling encryption and pixel grayscale value encryption. Scrambling encryption refers to using algorithms to change the position of pixels. Scrambling encryption can reduce the correlation between pixels, but it cannot change the pixel values. Amnesh et al. proposed a scrambling encryption scheme based on RGB translation scrambling to achieve the encryption of colored images [3]. Jolfaei and Mirghadri proposed a scrambling encryption scheme based on the Henon chaotic map [4]. Alexopoulos proposed a scrambling encryption method based on the SCAN mode [5]. This scheme used different routes to scan the image to be encrypted. Pixel grayscale value encryption refers to changing the pixel grayscale values to achieve encryption. El-Zoghdy et al. used the DES algorithm to encrypt images [6]. Acharya et al. proposed a scheme based on Hill matrix encryption algorithm and used an invertible matrix to encrypt images [7]. With increasing research on image encryption algorithms, some scholars have proposed many hybrid encryption schemes. In 2019, we have proposed a chaos-based image encryption technique utilizing Hilbert Curves and H-Fractals [8].

Data encryption standard (DES) was established by the US Federal Government in 1977. It has been authorized and

applied to governmental nonconfidential communications. The traditional DES algorithm [9] first groups the plaintext into many blocks, and the size of each block is 64 bits. Then, each block is divided into two parts, which are scrambled and diffused for 16 rounds. Finally, the encrypted blocks are rearranged. Encrypting images with the traditional DES algorithm faces many problems, such as complicated operation, poor encryption effect, and weak antiocclusion attack capability. For example, Silva-Garcia et al. noticed that when the traditional DES algorithm was applied to images that had large blocks of the same grayscale value [10], some regions with low encryption quality were generated. Thus, the traditional DES algorithm is not suitable for image encryption.

As a complex nonlinear dynamic system [11–14], the chaotic system has the characteristics of high initial parameter sensitivity, unpredictable orbit, and strong state ergodicity. It is usually used as a pseudorandom number generator. Applying the chaotic system to the scrambling encryption process and pixel grayscale value, encryption process can improve the security of the encryption system and overcome the shortcomings of the traditional DES algorithm. Considering this theory, a new image encryption algorithm based on the traditional DES encryption algorithm model, chaotic systems, DNA computing, and select cipher-text output is proposed. The simulation results and security analysis verified the feasibility of this algorithm.

2. Fundamental Theory

2.1. Chaotic System. To increase the antiocclusion capability of this encryption scheme, the logistic map is chosen to scramble the position of pixels [15]. The logistic map is defined in formula (1). In this equation, μ is the parameter; when μ is 4, the chaotic system is in the complete chaos state. The phase diagram of the logistic map is shown in Figure 1(a).

$$a(i+1) = \mu a(i)(1 - a(i)). \tag{1}$$

To reduce the correlation between adjacent pixels, the 2D-LSCM chaotic system is chosen to encrypt the pixel grayscale value [16]. The 2D-LSCM chaotic system is defined as formula (2). In this equation, θ is the parameter of the chaotic system. When θ is in the interval (0, 1), the chaotic system is in the chaos state. The phase diagram of the 2D-LSCM chaotic system is shown in Figure 1(b).

$$\begin{cases} x(i+1) = \sin(\pi(4\theta x(i)(1 - x(i)) + (1 - \theta)\sin(\pi y(i)))), \\ y(i+1) = \sin(\pi(4\theta y(i)(1 - y(i)) + (1 - \theta)\sin(\pi x(i+1)))). \end{cases} \tag{2}$$

2.2. DNA Coding and Nucleotide Computing. In 1994, Adleman designed and completed the first DNA computing experiment and published his experimental results in the journal of Science [17], opening a new field, biocomputing. Many scholars have performed various studies on biocomputing. Derived from the structure of DNA, DNA computing has many excellent features, such as huge storage capacity, large parallel computing power, and ultralow power consumption. The field of biocomputing is still being explored and studied continuously. Gehani et al. proposed the first algorithm to encrypt images using DNA sequences [18]. In 2012, Chang proposed a fast parallel DNA-based algorithm. This algorithm uses the RSA public-key cryptosystem, biological cryptography, and biological parallel computing to encrypt images [19]. Although DNA computers have not been applied in practice, the encoding and computing methods of DNA computing enrich the means of encryption algorithms in cryptography. By simulating DNA calculations, various evaluation indexes of the encryption algorithm are improved [20, 21].

2.2.1. DNA Coding. Double-stranded DNA molecules consist of four nucleotides: A (adenine), T (thymine), G (guanine), and C (cytosine). Nucleotides follow the principles of complementary pairing, namely, A and T are complementary, and G and C are complementary. In DNA coding, each nucleotide can represent a 2-bit binary string. Pixels in the grayscale images are in the interval [0, 255], so they can be represented by the four nucleotides [22–24]. In binary numbers, 0 and 1 are complementary. Therefore, 00 and 11 are complementary, and 01 and 10 are complementary. When an 8-bit binary string is represented by the four nucleotides, there are $4! = 24$ encoding rules, but only 8 kinds of rules satisfy the principle of complementary pairing. These rules are listed in Table 1.

When converting pixels to DNA coding, we first need to determine the rule for encoding. For example, the binary string of the decimal digit 188 is 10111100, which can be represented by the nucleotide sequence CTTA under rule 1. DNA decoding is the reverse process of DNA encoding. A different digit is obtained when decoding nucleotide sequences under a different rule. For example, if we decode the nucleotide sequence CTTA under rule 2, we will obtain a binary string 01111100, where the decimal digit of this binary string is 124, which is different from 188. In the encryption process, using different rules to encode and decode is an encryption scheme, but in the decryption process, we should ensure the consistency of encoding and decoding rules.

2.2.2. Nucleotide Computing. With the advances in biocomputing research, some scholars have proposed algebraic operations based on nucleotides [25], such as nucleotide addition and subtraction. Nucleotide addition and subtraction are variants of binary addition and subtraction. There are 8 kinds of nucleotide addition rules, and 8 kinds of nucleotide subtraction rules corresponding to 8 DNA coding rules. The nucleotide addition and subtraction rules under the first coding rule are shown in Table 2.

The nucleotide operations are adding or subtracting binary digits represented by nucleotides. Unlike binary addition or subtraction, only the last 2-bit binary digits remain for the results. For example, the nucleotide sequences TCAG and GATC are added under the first

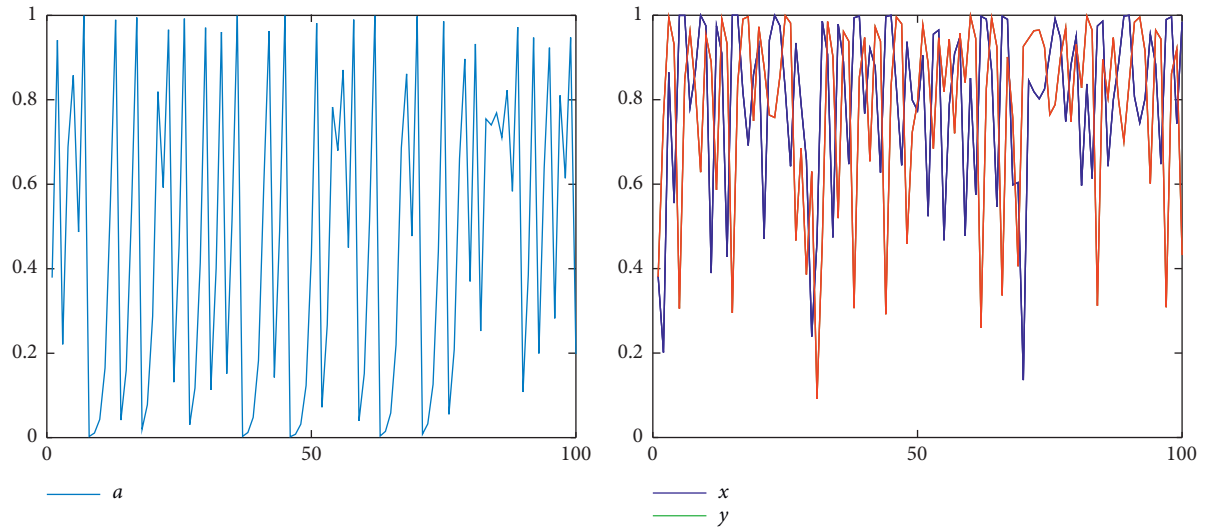

Figure 1: Simulation figure of chaotic systems. The phase diagram of (a) logistic map and (b) 2D-LSCM system.

Table 1: 8 kinds of DNA coding rules.

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

coding rule, and the result is ACTT. The sequence ACTT is used to reduce the sequence of GATC under the first rule, and the result is TCAG. The operation processes are shown in Figure 2. In the addition and subtraction operation processes, the operation results corresponding to each rule are unique.

3. Design of the Encryption Scheme

To improve the security of the key, the encryption scheme uses the SHA-256 algorithm to operate the plaintext image and obtain a 256-bit hash sequence as an initial key, so there is a connection between the key and original image. To improve the antiocclusion capability of the encryption system, this encryption scheme uses the logistic map to globally scramble the pixels. Finally, the encryption scheme uses block diffusion, forward diffusion, and entropy selection methods to improve the evaluation indexes of the encryption system to resist statistical attacks and enhance the pseudorandomness of the encryption system. The details of the encryption process are shown as follows.

3.1. Key Generator. SHA-256 is a type of hash algorithm [26, 27] that can convert any length of data into a 256-bit binary sequence. The hash sequences generated by the SHA-256 algorithm are irreversible, so the plaintext information cannot be inversely calculated from the hash sequences. In this paper, the 256-bit binary sequence H is generated by the SHA-256 algorithm as the initial key. To calculate the initial values of the logistic map and the 2D-LSCM chaotic system, the hash sequence is divided into 32 equal-sized parts $k_1, k_2, \ldots, k_{32}$. They are integers in the interval [0, 255] when they are converted into decimal digits. Then, the initial values are obtained through formula (3). Pseudorandom sequences are calculated through chaotic systems.

$$\begin{cases} \text{init}_1 = \dfrac{(k_1 \oplus k_2 \oplus k_3 \oplus k_4 \oplus k_5 \oplus k_6 \oplus k_7 \oplus k_8)}{256}, \\ \text{init}_2 = \dfrac{(k_9 \oplus k_{10} \oplus k_{11} \oplus k_{12} \oplus k_{13} \oplus k_{14} \oplus k_{15} \oplus k_{16})}{256}, \\ \text{init}_3 = \dfrac{(k_{17} \oplus k_{18} \oplus k_{19} \oplus k_{20} \oplus k_{21} \oplus k_{22} \oplus k_{23} \oplus k_{24})}{256}, \\ \text{init}_4 = \dfrac{(k_{25} \oplus k_{26} \oplus k_{27} \oplus k_{28} \oplus k_{29} \oplus k_{30} \oplus k_{31} \oplus k_{32})}{256}, \\ \text{init}_5 = k_{25} \oplus k_{26} \oplus k_{27} \oplus k_{28} \oplus k_{29} \oplus k_{30} \oplus k_{31} \oplus k_{32}, \\ a(1) = \text{init}_4, \\ x(1) = \dfrac{(\text{init}_1 + \text{init}_5)}{256}, \\ y(1) = \dfrac{(\text{init}_2 + \text{init}_5)}{256}, \\ \theta = \dfrac{(\text{init}_3 + \text{init}_5)}{256}. \end{cases} \tag{3}$$

Table 2: Nucleotide computing rules under the first coding rule.

+	**A**	**G**	**C**	**T**	−	**A**	**G**	**C**	**T**
A	A	G	C	T	A	A	T	C	G
G	G	C	T	A	G	G	A	T	C
C	C	T	A	G	C	C	G	A	T
T	T	A	G	C	T	T	C	G	A

TCAG 11 10 00 01
+ GATC 01 00 11 10
= ACTT 00 10 11 11
(a)

ACTT 00 10 11 11
− GATC 01 00 11 10
= TCAG 11 10 00 01
(b)

Figure 2: Examples of nucleotide operations under the first coding rule.

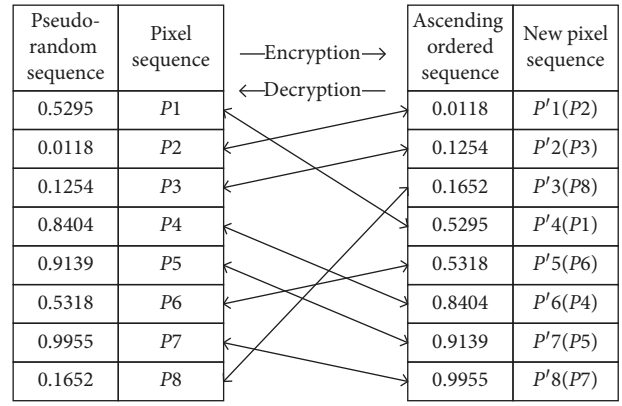

Figure 3: Scrambling process and its decryption process.

3.2. Pixel Position Scramble. Scrambling is a method of changing pixel locations [28, 29]. In this paper, to reduce the correlation between adjacent pixels, the logistic map is applied to scramble the position of pixels. Scrambling has three steps. The first step is mapping the positions of the elements in the pseudorandom sequence one by one with the positions of the pixels in the pixel sequence. The second step is arranging the elements in the pseudorandom sequence in ascending order. The third step is changing the position of the elements in the pixel sequence following the position of the elements in the pseudorandom sequence. We only need to obtain the original pseudorandom sequence and the scrambled pixel sequence to complete the decryption process because the decryption process is the reverse process of the scrambling process. Details of the scrambling and decryption processes are shown in Figure 3. To encrypt the entire image, first, we should use the logistic map to generate two matrices A_1 and A_2 which are the same size as the original image. Second, we should use every row of matrix A_1 to scramble every row of the original image. Third, we should use every column of matrix A_2 to scramble every column of the image whose rows are scrambled.

3.3. Grayscale Encryption

3.3.1. DES Diffusion. To increase the randomness of the cipher images, we adopt a block diffusion scheme similar to the DES structure. This diffusion scheme ignores the grouping operation, and it directly divides the images to be encrypted into two equal-sized matrices as left matrix L_0 and right matrix R_0. Then, we use a 2D-LSCM chaotic system to generate sixteen matrices with the same size of L_0 and R_0 as $K_0, K_1, \ldots, K_{16}$. Finally, matrices L_0 and R_0 are operated with matrices $K_0, K_1, \ldots, K_{16}$ through formula (4). We will reserve the matrices $L_{13}, \ldots, L_{16}$ and $R_{13}, \ldots, R_{16}$ for other purposes.

$$\begin{cases} L_i = R_{i-1}, R_i = \text{bitxor}\left(L_{i-1}, \text{bitxor}\left(R_{i-1}, K_i\right)\right), & 1 \le i \le 8, \\ L_i = R_{i-1}, R_i = \text{nucleotides subtraction} \\ \quad \left(L_{i-1}, \text{nucleotides addition}\left(R_{i-1}, K_i\right)\right), & 9 \le i \le 16. \end{cases} \tag{4}$$

3.3.2. Forward Diffusion. The purpose of diffusion is to correlate adjacent pixels [30–33]. Using forward diffusion to the encrypt image with the size of $M \times N$, we first rearrange the pixel matrix into a pixel sequence of length $M \times N$ and label the pixel points as $P(1), P(2), \ldots, P(M \times N)$. Then, we calculate the encrypted pixels $P'(1), P'(2), \ldots, P'(M \times N)$ through formula (5). Finally, we rearrange the pixel sequence into a matrix of size $M \times N$ to obtain the encrypted image.

$$\begin{cases} P'(1) = P(1), \\ P'(i) = \text{bitxor}\left(P(i), P'(i-1)\right), \quad i \ge 2. \end{cases} \tag{5}$$

Using forward diffusion can increase the information entropy of the image and reduce the correlation between adjacent pixels. The Lena image and forward diffused Lena image are shown in Figure 4. It can be seen from the comparison that the diffused image still retains some features of the original image, so the encryption algorithm cannot only use forward diffusion. The forward diffusion must be used in conjunction with other methods.

3.4. Encryption Scheme. The flow chart of the encryption scheme is shown in Figure 5. The 256×256 Lena image is encrypted as an example. Suppose that the original image is I, and the encryption scheme is as follows:

Step 1. The original image I is input into the SHA-256 algorithm to obtain a 256-bit binary hash sequence H.

Step 2. The hash sequence H is divided into 32 equal parts as $k_1, k_2, \ldots, k_{32}$; then initial values $a(1)$, $x(1)$, $y(1)$, θ of the logistic map and the 2D-LSCM chaotic system are calculated using formula (3).

Step 3. Generate two matrices, A_1 and A_2 of size 256×256, by iterating and reshaping the logistic map. Generate sixteen matrices, K_1 to K_{16} of size 256×128, by iterating and reshaping the 2D-LSCM system. To make the system full divergence, abandon the previous 1000 elements of each sequence in the chaotic systems.

(a) (b)

Figure 4: Effect of forward diffusion. (a) Lena image. (b) Forward diffused Lena image.

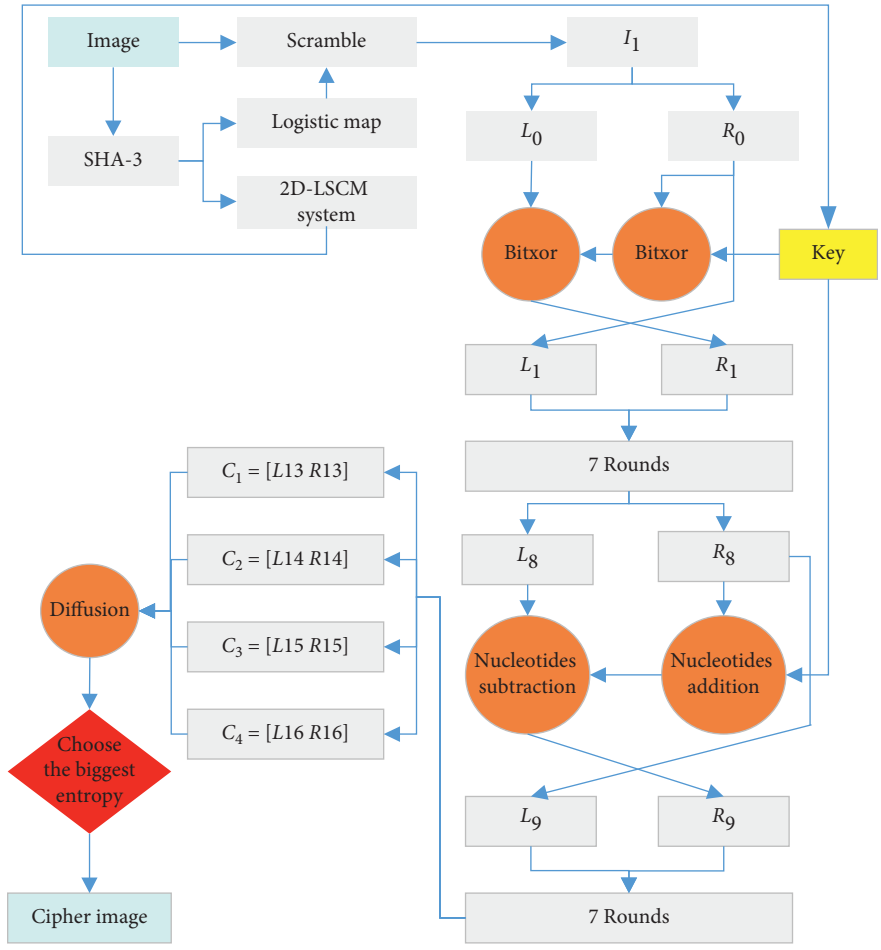

Figure 5: Flow chart of the encryption process.

Step 4. Image I_1 is obtained using the scrambling method of Section 3.2 with A_1 and A_2.

Step 5. Image I_1 is divided into two equal parts, L_0 and R_0, and the block diffusion operation is carried out by substituting formula (4). The ciphers L_{13} and R_{13} after 13 rounds of block diffusion make cipher image C_1, the ciphers L_{14} and R_{14} after 14 rounds of block diffusion make cipher image C_2, the ciphers L_{15} and R_{15} after 15 rounds of block diffusion make cipher image C_3, and the ciphers L_{16} and R_{16} after 16 rounds of block diffusion make cipher image C_4.

Step 6. The forward diffusion method of Section 3.3 is used to diffuse the images of C_1, C_2, C_3, and C_4, and new images C_1', C_2', C_3', and C_4' are obtained.

Step 7. Choose the image with the largest information entropy in C_1', C_2', C_3', and C_4' as the output of cipher image C.

3.5. Decryption Process. The process of the encryption algorithm proposed in this paper is reversible, but because it is uncertain which round of cipher the image C is generated, the collision process needs to be added in the decryption process. The steps of the decryption process are summarized as follows:

Step 1. Decrypt cipher image C by forward diffusion decryption operations as C_1, C_2, C_3, and C_4. Decrypt DES and scramble decryption operations through the

reverse process of encryption process to obtain decrypted images I_1, I_2, I_3, and I_4. Because the encryption process is reversible, the decryption process will not be repeated.

Step 2. The hash values H_1, H_2, H_3, and H_4 of images I_1, I_2, I_3, and I_4 were calculated by the SHA-256 algorithm, and the information entropies E_1, E_2, E_3, and E_4 of images I_1, I_2, I_3, and I_4 were calculated.

Step 3. Compare H_1, H_2, H_3, and H_4 with the initial key H, which is equal to H, which is that of the decrypted image. If H_1, H_2, H_3, and H_4 are not equal to the initial key H, compare with entropies E_1, E_2, E_3, and E_4, and the image with the smallest information entropy is selected as the output of the decrypted image. At the same time, a hint is given that there is a deviation between the decrypted image and the original image.

4. Simulation Results and Security Analysis

Some common images were used to verify the feasibility and security of the encryption algorithm. The original images, the encrypted images, and the decrypted images are shown in Figure 6. In our simulation, the cipher images have completely lost the characteristics of the original images. After decrypting the cipher images, each decrypted image is exactly the same as the original image, so the algorithm is lossless.

4.1. Key Sensitivity Analysis

4.1.1. Key Space Analysis. A good encryption scheme should have enough key space to resist a brute force attack. The keys used in this paper include the hash sequence $a(1)$, $x(1)$, $y(1)$, and θ. The key space of the SHA-256 algorithm is 2^{128}, and the precision of the initial value is calculated by 10^{-15}. Then, the total key space of the SHA-256 algorithm is $2128 * 1015 * 1015 * 1015 * 1015 = 3.4028 * 1098$. Thus, the algorithm has enough key space to resist a brute force attack and has strong security.

4.1.2. Key Sensitivity Analysis. A good encryption scheme should be sensitive to the key. The decrypted image obtained by changing one key by 10^{-15} with the other keys unchanged is shown in Figure 7; the encryption algorithm cannot be cracked. By comparison, it can be concluded that the encryption scheme is very sensitive to the key.

4.2. Antistatistical Attack Capability Analysis. The histogram of the image and the correlation between adjacent pixels are used to characterize the image. The feature of the original image is obvious, and the pixel values in some blocks of the image are distributed in concentration. When this phenomenon is reflected in the histogram, the elements are unevenly distributed. When this phenomenon is reflected in the correlation, the correlation between adjacent pixels is very strong. A good encryption algorithm can break the distribution characteristics of pixels in the original image, make the distribution of pixels in the cipher image more uniform, and reduce the correlation between adjacent pixels in the cipher image. Thus, attackers cannot attack cipher images by statistical means, and the encryption algorithm can effectively resist statistical attacks. In this paper, the histograms of original images and cipher images and the correlation between adjacent pixels are listed to show the ability of the algorithm to resist statistical attacks. At the same time, we add some comparisons with other image encryption algorithms to prove the advantages of the image encryption algorithm.

4.2.1. Histogram Analysis. In Figure 8, some histograms of the original images and histograms of the corresponding encrypted images are listed. In the histograms of the plaintext images, the pixel values are not uniform; it has certain statistical characteristics and cannot resist brute force attacks. However, in the cipher images, the distributions of pixel values are very uniform, and there is no statistical rule. The comparison between the histogram of the original image and the histogram of the encrypted image proves that the encrypted algorithm can break the statistical rule of the original images and has a good antistatistical attack ability.

4.2.2. Correlation Analysis. The correlation between adjacent pixels of the plaintext image is very strong. Breaking the correlation between adjacent pixels can enhance the ability of the encryption algorithm to resist statistical attacks. We randomly selected 10000 pixels from the original Lena image and the encrypted Lena image in the horizontal, vertical, and diagonal directions and listed these pixel values and their adjacent pixel values in Figure 9. There are strong correlations between adjacent pixels in the original Lena image, but there are almost no correlations between adjacent pixels in the encrypted Lena image. We can quantify the correlation between adjacent pixels with mathematical indicators. The correlation coefficient between adjacent pixels is calculated using formula (6), where $E(x)$ is the mean, $D(x)$ is the variance, and $\operatorname{cov}(x, y)$ is the covariance. The correlation coefficients are shown in Table 3. By comparing the correlation coefficients of the original images and cipher images in Table 3, it can be seen that the algorithm can resist statistical attacks very well, and its ability is not weaker than those of references [34] and [35].

$$\begin{cases} E(x) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} x_i, \\ D(x) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \left(x_i - E(x)\right)^2, \\ \operatorname{cov}(x, y) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \left(x_i - E(x)\right)\left(y_i - E(y)\right), \\ r_{xy} = \dfrac{\operatorname{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}. \end{cases} \tag{6}$$

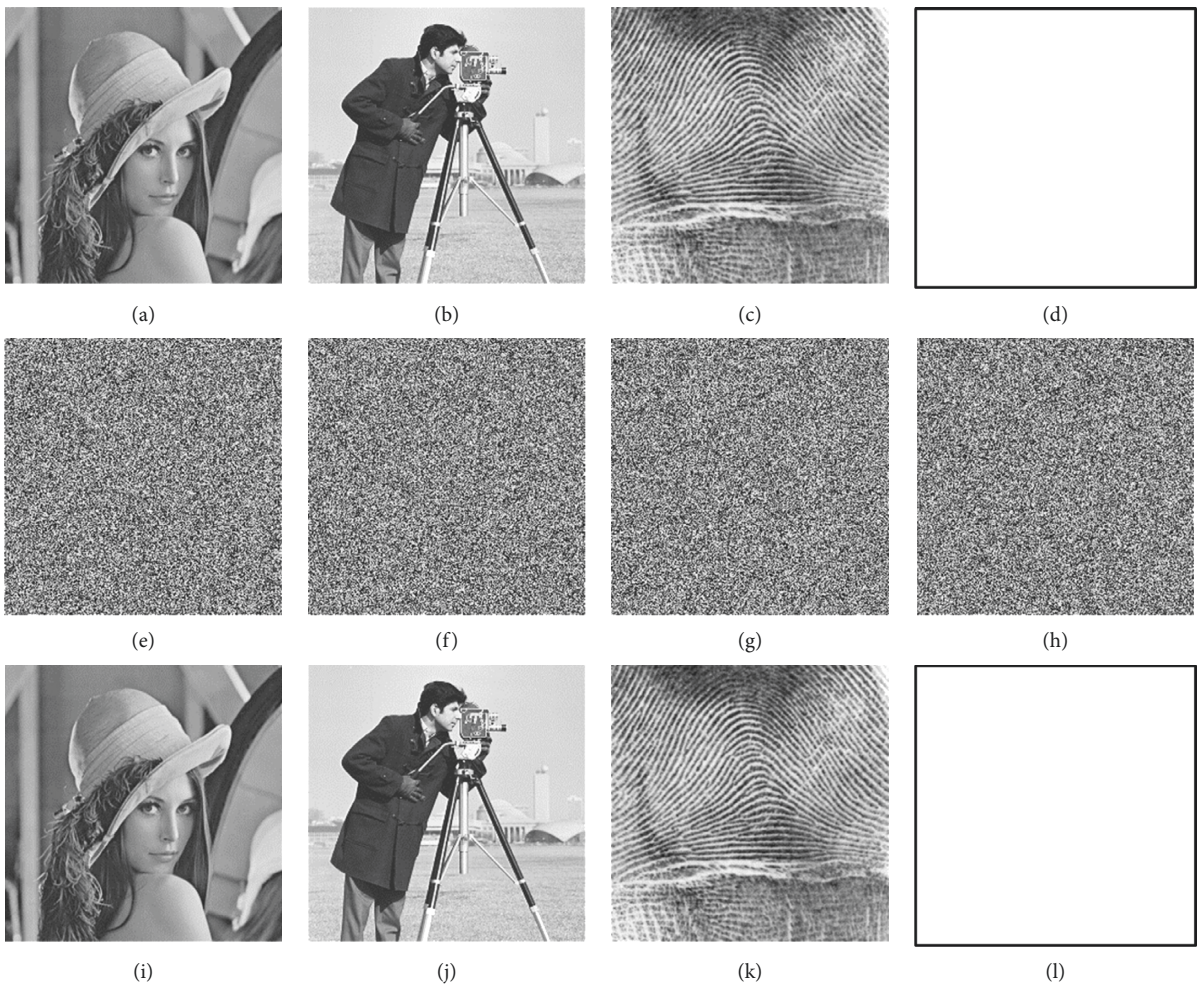

Figure 6: Original images, encrypted images, and decrypted images. (a) Lena. (b) Cameraman. (c) Fingerprint. (d) All white. (e) Encrypted Lena image. (f) Encrypted cameraman image. (g) Encrypted fingerprint image. (h) Encrypted all white image. (i) Decrypted Lena image. (j) Decrypted cameraman image. (k) Decrypted fingerprint image. (l) Decrypted all white image.

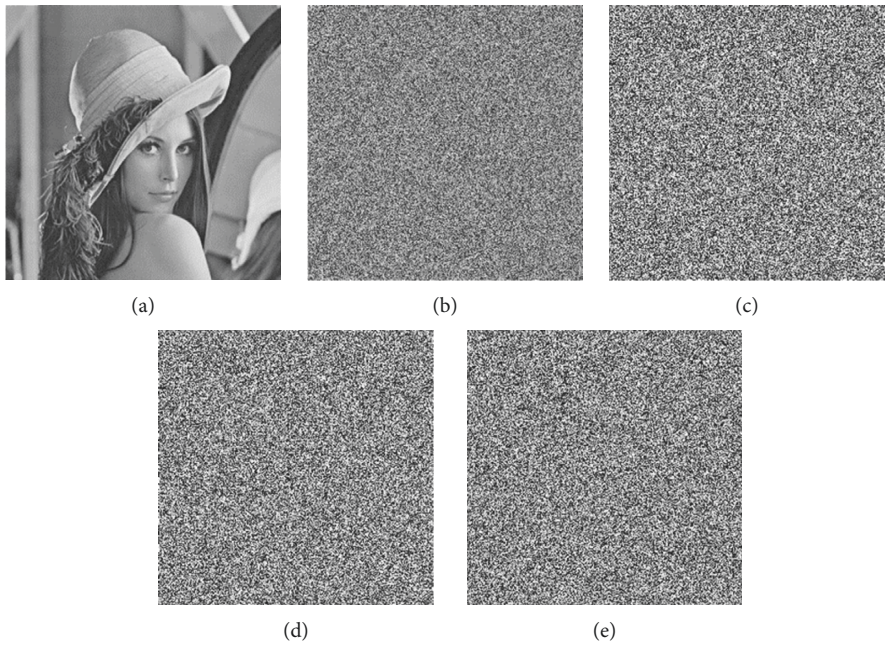

Figure 7: Decrypted images with the keys have minimal change. (a) Correct decrypted image. Decrypted image (b) when $a(1)$ changed 10^{-15}, (c) when $x(1)$ changed 10^{-15}, (d) when $y(1)$ changed 10^{-15}, and (e) when θ changed 10^{-15}.

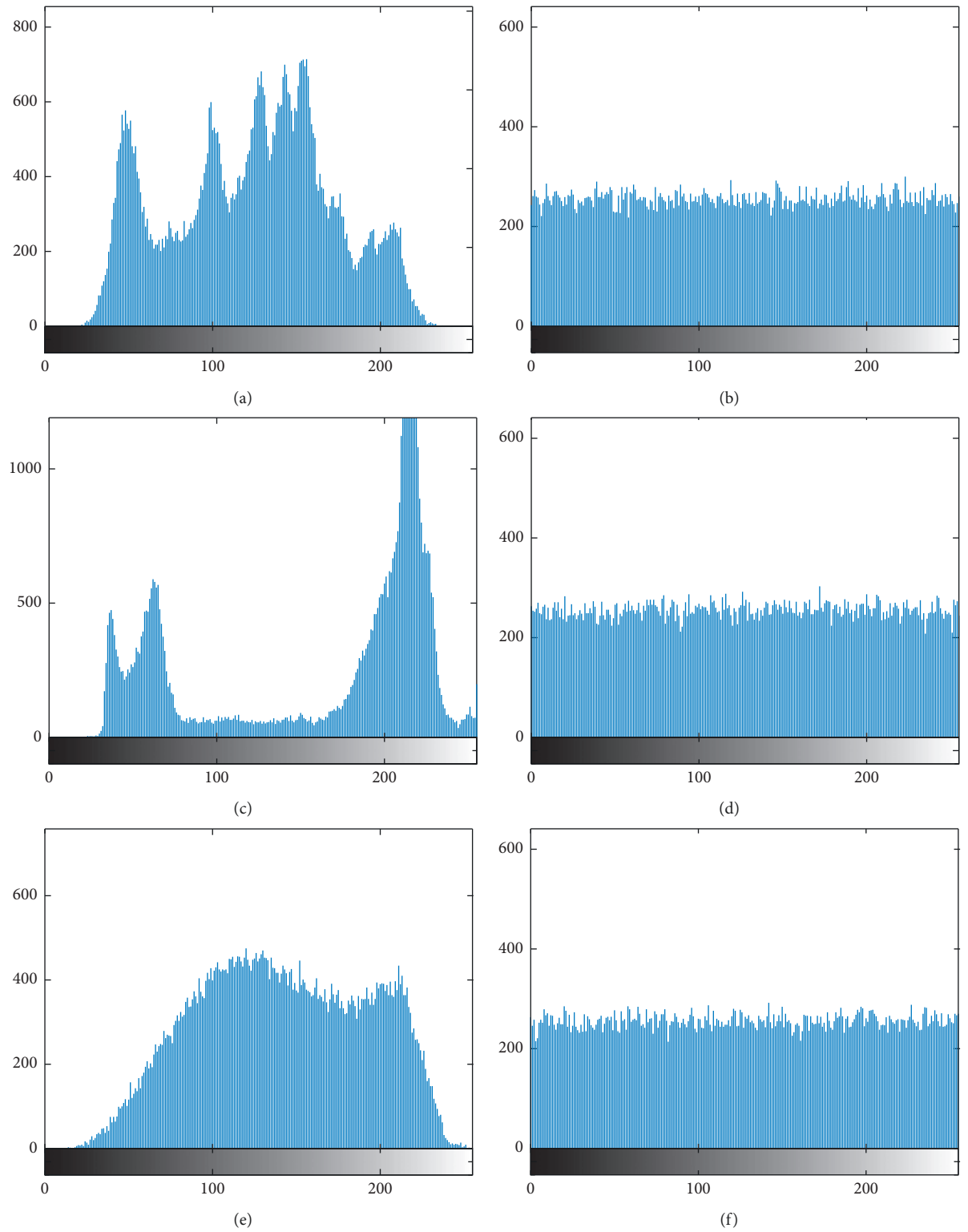

Figure 8: Histograms of original images and cipher images: (a) Lena image, (b) encrypted Lena image, (c) cameraman image, (d) encrypted cameraman image, (e) fingerprint image, and (f) encrypted fingerprint image.

4.2.3. Information Entropy Analysis. In 1948, Shannon put forward the concept of information entropy. The concept of information entropy solves the problem of quantifying and measuring information and can be used to judge the randomness of information. When the information entropy of a piece of information is close to its ideal value, we can judge

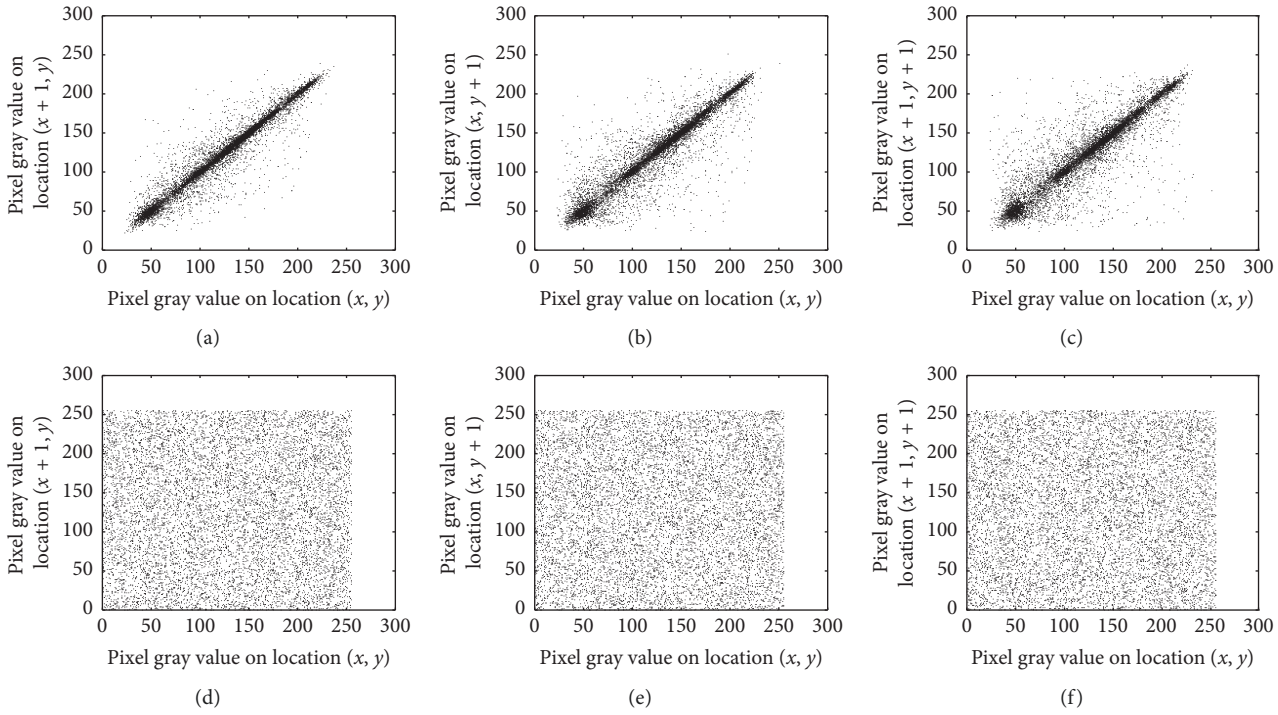

Figure 9: Correlations of adjacent pixels: (a) horizontal, (b) vertical, and (c) diagonal correlations of original image; (d) horizontal, (e) vertical, and (f) diagonal correlations of decrypted image.

Table 3: Correlations of original images and encrypted images.

Schemes	Images	Original image			Encrypted image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
This study	Lena	0.9677	0.9358	0.9020	0.0048	0.0001	0.0018
	Cameraman	0.9543	0.9131	0.8984	−0.0055	0.0048	−0.0011
	Fingerprint	0.7739	0.8153	0.6151	−0.0006	−0.0026	−0.0020
Ref. [34]	Lena	—	—	—	−0.0796	0.0166	0.0032
	Cameraman	—	—	—	−0.0398	−0.0387	−0.0090
Ref. [35]	Lena	0.9721	0.9739	0.9705	−0.0029	−0.0017	0.0004
	Cameraman	0.9634	0.9732	0.9449	0.0047	−0.0066	0.0031

that the information has good randomness. The information entropy of the image can be used to measure the degree of randomness of the image. The calculation method of information entropy is shown in formula (7), where $p(i)$ represents the probability that each situation may occur in a model:

$$H(s) = -\sum_{i=1}^{n} p(i)\log_2 p(i), \tag{7}$$

Pixel values are distributed in the interval [0, 255], and the probability of each case is 1/256, so the information entropy of a gray image is 8 in an ideal case. If the information entropy of a grayscale image is close to 8, the image has good randomness. The information entropies of the cipher images encrypted by this algorithm and some other algorithms are shown in Table 4. The information entropy of the cipher image of this algorithm is close to 8, and the result is not inferior to other algorithms. Therefore, it can be considered that the randomness of the cipher image of this algorithm meets the encryption requirements.

4.3. Antidifferential Attack Capability Analysis. The number of pixel change rate (NPCR) and unified average changing intensity (UACI) are two indices used to measure the correlation degree between a cipher image and an original image as well as the antidifferential attack ability of the encryption algorithm. The closer the NPCR and UACI are to the ideal values, the stronger the antidifferential attack ability of the encryption algorithm. The calculation methods of NPCR and UACI are shown in formula (8):

Table 4: Entropy of images.

	Lena	Cameraman	Fingerprint	All white
Original image	7.4532	6.9046	7.5945	0
Cipher image	7.9976	7.9974	7.9977	7.9979
Ref. [35]	7.9971	7.9971	7.9970	7.9970
Ref. [36]	7.9965	7.9964	7.9962	—
Ref. [37]	7.9968	7.9904	—	—

Table 5: NPCR and UACI.

Schemes	Images	NPCR (%)	UACI (%)
This study	Lena	99.5987	33.5501
	Cameraman	99.6231	33.5269
	Fingerprint	99.6597	33.5613
Ref. [34]	Lena	99.6521	33.3438
	Cameraman	99.6292	33.4140
Ref. [37]	Lena	99.58	33.08
	Cameraman	99.60	33.15
Ref. [38]	Lena	99.6078	28.6203

$$\begin{cases} \text{NPCR} = \dfrac{\sum_{i,j}\left|\text{Sign}\left(P_1(i,j) - P_2(i,j)\right)\right|}{M \times N} \times 100\%, \\ \text{UACI} = \dfrac{\sum_{i,j}\left|P_1(i,j) - P_2(i,j)\right|}{255 \times M \times N} \times 100\%, \end{cases} \tag{8}$$

$$\text{Sign}(x) = \begin{cases} 1, & x > 0, \\ 0, & x = 0, \\ -1, & x < 0. \end{cases} \tag{9}$$

$\text{Sign}(x)$ is a symbolic function, and its calculation method is shown in formula (9). $P_1(i, j)$ is the pixel value of the cipher image, and $P_2(i, j)$ is the pixel value of the cipher-text image encrypted after a slight change in the original image. The theoretical expectations of NPCR and UACI were 100% and 33.4635%, respectively. The NPCR and UACI values of the two cipher images are shown in Table 5. By comparison, we can see that the cipher image of this algorithm has a strong correlation with the original image, and it can resist the differential attack very well.

4.4. Antinoise Attack Capability Analysis. The antinoise attack ability of an encryption system is one of the standards for measuring the robustness of the encryption system. In the process of transmission, information will inevitably be disturbed by noise, which will distort the cipher image and affect the decrypted image. The common noises are Gauss noise, Poisson noise, salt and pepper noise, etc. In this section, the antinoise attack ability of the encryption system is analyzed. Different intensities of salt and pepper noise are added to the cipher image using MATLAB software and then decrypted. The simulation results are shown in Figure 10. The correlation of the image is used as an index to compare the decrypted image with the original image. The correlations are shown in Table 6. Based on this analysis, the encryption algorithm has a good antinoise attack ability.

4.5. Antiocclusion Attack Capability Analysis. The anti-occlusion attack ability of the encryption algorithm can reflect the scattered degree of cipher text. If the scrambling degree of the encryption algorithm is insufficient, the occlusion area in the decrypted image may completely lose its original characteristics after decryption. The occlusion attack test is to occlude the cipher image and then observe the degree of restoration of the decrypted image. The clipped cipher images are shown in Figures 11(a)–11(d). The cipher images with cutting areas of 1/64, 1/16, 1/4, and 1/2 are decrypted. The results are shown in Figures 11(e)–11(h). With the correlation of the image as an indicator, the decrypted image and the original image were compared and analyzed. The results are shown in Table 7. By analyzing the correlation, we can see that when the cipher image is attacked by clipping, the algorithm can restore the original image features to some extent. Therefore, the algorithm has a good antiocclusion attack ability.

5. Conclusions

In this paper, an image encryption algorithm based on block diffusion and the chaotic system is proposed by means of a traditional DES encryption algorithm combined with chaotic system and DNA coding technology. This method compensates for the problems of high computational complexity and inconvenient key management when the traditional text encryption algorithm is used in the digital image encryption algorithm by using DNA coding operation, selecting cipher-text output, and key verification. The experimental results show that the algorithm has a large key space to resist statistical attacks,

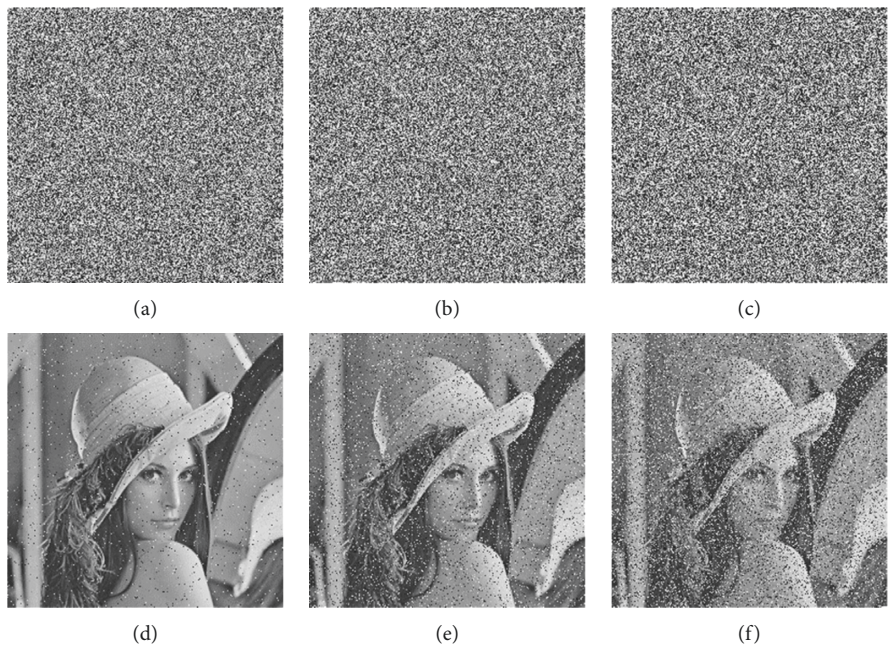

Figure 10: Decrypted images with different intensities of noise attacks. Noise intensity (a) of 0.01, (b) 0.04, and (c) 0.1; decrypted image with (d) 0.01 intensity noise, (e) 0.04 intensity noise, and (f) 0.1 intensity noise.

Table 6: Correlations of decrypted images with different intensities of noise attack.

Noise intensity	Correlations		
	Horizontal	Vertical	Diagonal
Original image	0.9677	0.9358	0.9020
0.01	0.8584	0.8214	0.7983
0.04	0.6262	0.5934	0.5714
0.1	0.3303	0.2831	0.2840

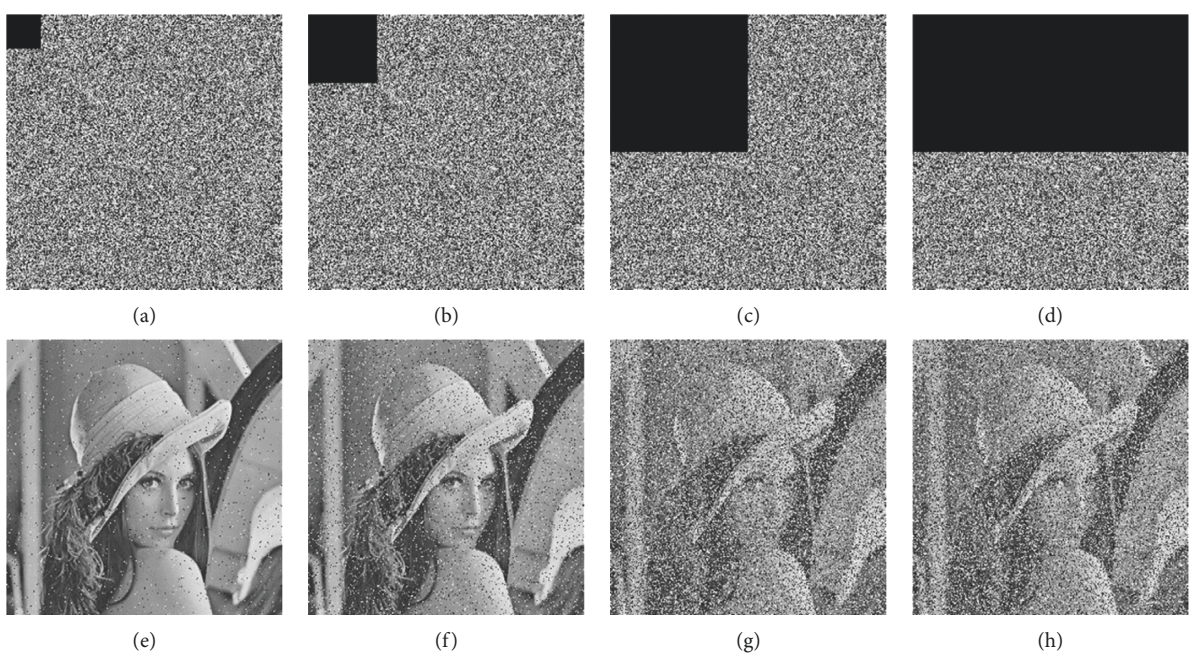

Figure 11: Decrypted images with occlusion attack. (a) Occlusion 1/64. (b) Occlusion 1/16. (c) Occlusion 1/4. (d) Occlusion 1/2. (e) Decrypted image with 1/64 occlusion. (f) Decrypted image with 1/16 occlusion. (g) Decrypted image with 1/4 occlusion. (h) Decrypted image with 1/2 occlusion.

Table 7: Correlations of decrypted images with different intensities of noise attack.

Occlusion	Correlations		
	Horizontal	Vertical	Diagonal
Original image	0.9677	0.9358	0.9020
1/64	0.8688	0.8378	0.8161
1/16	0.6399	0.6007	0.5880
1/4	0.1430	0.1329	0.1202
1/2	0.1397	0.1457	0.1296

differential attacks, occlusion attacks, noise attacks, etc. It can be widely used for the secure transmission of image information.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China (grant nos. 61572446, 61602424, and U1804262), Key Scientific and Technological Project of Henan Province (grant nos. 174100510009 and 192102210134), and Key Scientific Research Projects of Henan High Educational Institution (18A510020).

References

[1] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[2] K. Zhu, Z. Lin, and Y. Ding, "A new RSA image encryption algorithm based on singular value decomposition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 1, article 1954002, 2019.

[3] G. Amnesh, M. Reji, and C. Nidhi, "Image encryption based on inter pixel displacement of RGB values inside custom slices," *Journal of Terramechanics*, vol. 48, no. 5, pp. 325–332, 2011.

[4] A. Jolfaei and A. Mirghadri, "An image encryption approach using chaos and stream cipher," *Journal of Theoretical and Applied Information Technology*, vol. 19, pp. 117–125, 2010.

[5] N. Bourbakis and C. Alexopoulos, "Picture data encryption using scan patterns," *Pattern Recognition*, vol. 25, no. 6, pp. 567–581, 1992.

[6] S. F. El-Zoghdy, Y. A. Nada, and A. A. Abdo, "How good is the DES algorithm in image ciphering?," *International Journal of Advanced Networking and Applications*, vol. 2, no. 5, pp. 796–803, 2011.

[7] B. Acharya, S. K. Panigrahy, S. K. Patra, and G. Panda, "Image encryption using advanced Hill cipher algorithm," *International Journal of Recent Trends in Engineering*, vol. 1, no. 1, pp. 663–667, 2009.

[8] X. Zhang, L. Wang, Z. Zhou, and Y. Niu, "A chaos-based image encryption technique utilizing hilbert curves and H-fractals," *IEEE Access*, vol. 7, pp. 74734–74746, 2019.

[9] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 17–28, 2002.

[10] V. M. Silva-García, R. Flores-Carapia, I. López-Yáñez, and C. Rentería-Márquez, "Image encryption based on the modified triple-DES cryptosystem," *International Mathematical Forum*, vol. 7, no. 59, pp. 2929–2942, 2012.

[11] G. Qi, S. Du, G. Chen, Z. Chen, and Z. Yuan, "On a four-dimensional chaotic system," *Chaos, Solitons and Fractals*, vol. 23, no. 5, pp. 1671–1682, 2005.

[12] J. Lü, G. Chen, and D. Cheng, "A new chaotic system and beyond: the generalized lorenz-like system," *International Journal of Bifurcation and Chaos*, vol. 14, no. 5, pp. 1507–1537, 2004.

[13] G.-H. Li, "Modified projective synchronization of chaotic system," *Chaos, Solitons and Fractals*, vol. 32, no. 5, pp. 1786–1790, 2007.

[14] Q. Zhang and J.-A. Lu, "Chaos synchronization of a new chaotic system via nonlinear control," *Chaos, Solitons and Fractals*, vol. 37, no. 1, pp. 175–179, 2008.

[15] A. H. Zhang and Z. Q. Jiang, "Improving for chaotic image encryption algorithm based on logistic mapping," *Journal of Nanjing University of Posts and Telecommunications*, vol. 3, pp. 211–214, 2009.

[16] Z. Hua, F. Jin, B. Xu, and H. Huang, "2D logistic-sine-coupling map for image encryption," *Signal Processing*, vol. 149, pp. 148–161, 2018.

[17] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, no. 5178, pp. 1021–1024, 1994.

[18] A. Gehani, T. Labeanv, and J. Reif, "DNA-based cryptography," in *Aspects of Molecular Computing*, vol. 54, no. 456, pp. 233–249, 2004.

[19] W.-L. Chang, "Fast parallel DNA-based algorithms for molecular computation: quadratic congruence and factoring integers," *IEEE Transactions on Nanobioscience*, vol. 11, no. 1, pp. 62–69, 2012.

[20] W.-L. Chang and A. V. Vasilakos, "DNA algorithms of implementing biomolecular databases on a biological computer," *IEEE Transactions on Nanobioscience*, vol. 14, no. 1, pp. 104–111, 2015.

[21] W.-L. Chang, A. V. Vasilakos, and M. S. Ho, "The DNA-based algorithms of implementing arithmetical operations of complex vectors on a biological computer," *IEEE Transactions on Nanobioscience*, vol. 14, no. 8, pp. 907–914, 2015.

[22] 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.

[23] Y. Liu, J. Tang, and T. Xie, "Cryptanalyzing a RGB image encryption algorithm based on DNA encoding and chaos map," *Optics and Laser Technology*, vol. 60, pp. 111–115, 2014.

[24] W.-L. Chang, T.-T. Ren, and M. Feng, "Quantum algorithms and mathematical formulations of bio-molecular solutions of vertex cover problem in the finite-dimensional hilbert space," *IEEE Transactions on Nanobioscience*, vol. 14, no. 1, pp. 121–128, 2015.

[25] R. Soni and A. Johar, "An encryption algorithm for image based on DNA sequence addition operation," *IEEE Transactions on Information Theory*, vol. 56, pp. 296–315, 2012.

[26] M. Fenwick, "Analysis of step-reduced SHA-256," in *Lecture Notes in Computer Science*, vol. 4047, no. 3, pp. 126–143, 2006.

[27] A. W. Appel, "Verification of a cryptographic primitive," *ACM Transactions on Programming Languages and Systems*, vol. 37, no. 2, pp. 1–31, 2015.
[28] Z.-L. Zhu, W. Zhang, K.-W. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Information Sciences*, vol. 181, no. 6, pp. 1171–1186, 2011.
[29] Y.-Q. Zhang and X.-Y. Wang, "Analysis and improvement of a chaos-based symmetric image encryption scheme using a bit-level permutation," *Nonlinear Dynamics*, vol. 77, no. 3, pp. 687–698, 2014.
[30] R. Ye, "A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism," *Optics Communications*, vol. 284, no. 22, pp. 5290–5298, 2011.
[31] C. Fu, J.-J. Chen, H. Zou, W.-H. Meng, Y.-F. Zhan, and Y.-W. Yu, "A chaos-based digital image encryption scheme with an improved diffusion strategy," *Optics Express*, vol. 20, no. 3, pp. 2363–2378, 2012.
[32] J.-X. Chen, Z.-L. Zhu, C. Fu, L.-B. Zhang, and Y. Zhang, "An efficient image encryption scheme using lookup table-based confusion and diffusion," *Nonlinear Dynamics*, vol. 81, no. 3, pp. 1151–1166, 2015.
[33] P. K. Sharma, M. Ahmad, and P. M. Khan, "Cryptanalysis of image encryption algorithm based on pixel shuffling and chaotic S-box transformation," *Communications in Computer and Information Science*, vol. 467, no. 7, pp. 173–181, 2014.
[34] R. Enayatifar, A. H. Abdullah, and I. F. Isnin, "Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence," *Optics and Lasers in Engineering*, vol. 56, pp. 83–93, 2014.
[35] C. Fu, G.-Y. Zhang, O. Bian, W.-M. Lei, and H.-F. Ma, "A novel medical image protection scheme using a 3-dimensional chaotic system," *PLoS One*, vol. 9, no. 12, Article ID e115773, 2014.
[36] X. Wang, X. Zhu, and Y. Zhang, "An image encryption algorithm based on Josephus traversing and mixed chaotic map," *IEEE Access*, vol. 6, pp. 23733–23746, 2018.
[37] H. Natiq, N. M. G. Al-Saidi, M. R. M. Said, and A. Kilicman, "A new hyperchaotic map and its application for image encryption," *European Physical Journal Plus*, vol. 133, no. 1, pp. 1–14, 2018.
[38] S. Mozaffari, "Parallel image encryption with bitplane decomposition and genetic algorithm," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 25799–25819, 2018.