JPEG Image Encryption Algorithm Based on Hyperchaotic, Mixed Hash and Dynamic DNA

Qiu-Yu Zhang and Yu-Tong Ye
(*Corresponding author: Qiu-Yu Zhang*)

School of Computer and communication, Lanzhou University of Technology
No. 287, Lan-Gong-Ping Road, Lanzhou 730050, China
Email: zhangqylz@163.com

(Received July 28, 2020; Revised and Accepted May 24, 2021; First Online Feb. 26, 2022)

Abstract

This paper proposes a new JPEG image encryption algorithm based on hyperchaotic, mixed hash, and dynamic DNA to solve the problem of poor robustness and low efficiency in existing JPEG image encryption schemes. First, the plaintext image is converted into a DNA matrix through DNA coding rules and permutated by the encryption key, generated by the chaotic system and a mixed hash function (MD5 and SHA-256). Secondly, the encryption key is used to dynamically control the DNA operation for diffusion. Finally, the ciphertext image is obtained by decoding the DNA codec rules. Experiments show that the proposed Algorithm is efficiently resistant to statistical attacks and has efficient encryption. Furthermore, the critical space can reach 2^{507}, the critical sensitivity is high, the pixel correlation coefficient is close to 0, the information entropy is close to 8, the UACI and NPCR values are close to the ideal value, and it has good robustness against noise, cropping and JPEG compression attacks.

Keywords: Dynamic DNA Coding; Image Encryption; JPEG Image; Mixed Hash; 5D Lorenz Hyperchaotic System

1 Introduction

With the rapid development of Internet technology, the speed and scale of data dissemination have reached unprecedented levels. The insecure factors in transmission and the image itself are characterized by large data volume and high redundancy makes the security and efficiency of digital image transmission extremely important [2, 23].JPEG images are widely used in daily life. With the widespread popularity of the network, the number of malicious programs such as viruses and Trojan horses is also increasing. Therefore, information security technologies such as JPEG image encryption and data hiding have received extensive attention [3, 6, 7, 9–11, 13, 14, 17–19].

At present, the number of encryption algorithms for specific image formats is still relatively small. Existing JPEG image encryption algorithms are mainly aimed at scrambling DC coefficients. For example, Siricottedumrong *et al.* [18] reduced the color information by dividing the image into smaller blocks to improve encryption performance of the scheme. To increase the algorithm resistance to statistical attacks. Chuman *et al.* [3] based on the puzzle algorithm proposed a new scheme. However, the scheme in [3, 18] inability to resist multiple types of attacks. Li *et al.* [10] combined compression and encryption algorithms to give two encryption schemes to improve compression performance and algorithm security. On the premise of ensuring compression efficiency, Li *et al.* [9] devised a scheme that effectively improved the encryption performance. He *et al.* [7] based on bitstream proposed an encryption scheme which is improved in the aspect of format compatibility and can effectively resist many types of attacks.

Chaos encryption technology is widely used, because of its pseudo-random characteristics and high sensitivity [6, 13]. Man *et al.* [13] devised an image segmentation encryption scheme. The efficiency of the key generation can be improved by using a set of chaotic sequences as the key tool to obtain the key. But the quantized AC coefficient value is significantly smaller than the quantized DC coefficient and the DC coefficient is easy to locate, so this scheme cannot effectively resist contour attacks. Ghazvini *et al.* [6] proposed a scheme by combining the advantages of genetic algorithms and chaotic systems. Use the scramble-diffusion framework to obtain the ciphertext image. The ciphertext image is obtained using a genetic algorithm to optimize the ciphertext image. However, chaotic systems used in these schemes have limitations. The practical application range of the encryption scheme will be limited. Mondal *et al.* [14] devised a secure image encryption scheme based on cellular automata and oblique tent mapping. This scheme is ideal in terms of robustness but poor in encryption efficiency. Combine with the chaotic system and permutation, Li *et*

al. [11] devised an encryption scheme. By summarizing the above encryption schemes, the existing methods have their advantages and disadvantages.

In recent years, hyperchaotic encryption systems have the advantage of high speed but have poor in key sensitivity. To increase the security of multimedia data and combine the advantages of various encryption technologies, more and more hybrid encryption schemes are proposed [17, 19]. Due to the high parallelism and high storage density of DNA molecules, DNA technology is widely used in the field of cryptography to improve the efficiency and security of encryption schemes. Salman *et al.* [17] proposed an encryption scheme based on DNA coding and chaotic systems. This scheme has better encryption performance and shorter encryption time for JPEG images. Thanikaiselvan *et al.* [19] proposed a two-stage reversible data hiding scheme based on DNA coding and chaotic systems. Steganographic images are encrypted by combining hash function, DNA coding and chaotic system.

Therefore, to overcome the deficiency of existing methods, this paper takes the JPEG image as the research carrier and presents a JPEG image encryption algorithm based on hyperchaotic, mixed hash and dynamic DNA. The main contributions of this paper can be summarized as follows:

1) SHA-256 is combined with MD5 hash function, and plaintext hash sequence is obtained by calculating the plaintext image, to increase the encryption systems for the chosen-plaintext attacks and known-plaintext attacks resistance.

2) DNA coding and operation rules are dynamically controlled by an encryption key, and the operation results cannot be accurately predicted, which improves the security of the key.

3) 5D Lorenz hyperchaotic system is used to expand the key space of the encryption algorithm, making the encryption algorithm more secure.

The remaining part of this paper is organized as follows. Section 2 describes the relevant theories in detail. Section 3 gives the JPEG image encryption scheme and its processing. Section 4 gives the experimental results and the performance analysis compared with other related methods. Finally, we conclude our work in Section 5.

2 Related Theories Analysis

2.1 5D Lorenz Hyperchaotic System

The classical Lorenz system dynamic equation [12] is shown in Equation (1):

$$\begin{cases} \frac{dx}{dt} = a(y-x) \\ \frac{dy}{dt} = bx - xz - y \\ \frac{dz}{dt} = xy - cz \end{cases} \tag{1}$$

where $a = 10$, $c = 8/3$, when $b > 24.74$,the Lorenz system enters into chaotic state.

The low-dimensional chaotic system has a simple structure and low algorithm complexity, but the key space is small and cannot effectively resist brute-force attacks. The control parameters, dynamic characteristics, and initial conditions of hyperchaotic systems are more complex than those of low-dimensional chaotic systems, which can expand the key space and possess stronger pseudo-randomness. Therefore, based on Equation (1), this paper constructs a 5D Lorenz hyperchaotic system by introducing two new variables u and w and nonlinear terms. The equation is shown in Equation (2):

$$\begin{cases} \frac{dx}{dt} = b(y-x) + yz - u + w \\ \frac{dy}{dt} = a(x+y) - xzu \\ \frac{dz}{dt} = -(c-a)z + xyu \\ \frac{du}{dt} = mu - xyz \\ \frac{dw}{dt} = -hx - hy \end{cases} \tag{2}$$

where x, y, z, u and w is the state variable of the constructed 5D Lorenz hyperchaotic system. a, b, c, m, and h are used to represent system parameters.

When $a = 27.02$, $b = 48$, $c = 33$, $m = 24$, $h = 9.48$, the Lyapunov exponents are $L_1 = 5.18340 > 0$, $L_2 = 3.00559 > 0$, $L_3 = 0.00054 \approx 0$, $L_4 = -17.74081 < 0$, $L_5 = -29.44874 < 0$,the chaotic system of Equation (2) is in a hyperchaotic state. Figure 1 shows the Lyapunov exponent diagram of 5D Lorenz hyperchaotic system. Figure 2 shows the chaotic attractor in each plane of the 5D Lorenz hyperchaotic system.

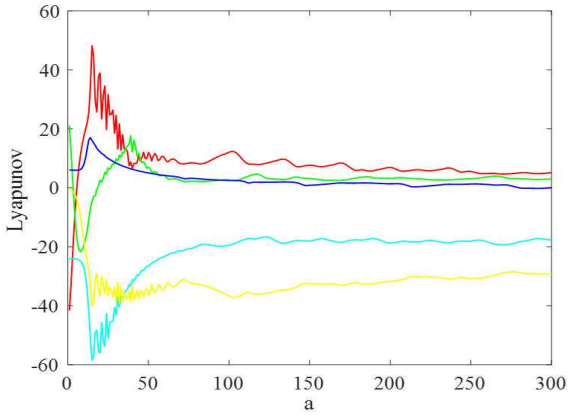
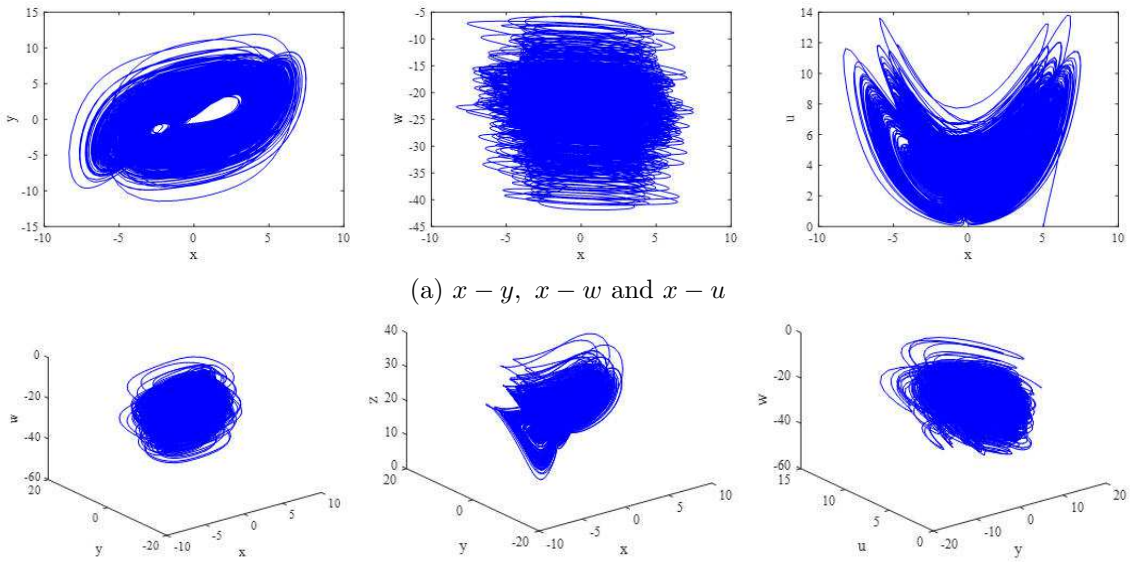


Figure 1: The Lyapunov exponent diagram of 5D Lorenz hyperchaotic system

2.2 Mixed Hash Function

During the process of image encryption, the hash function is usually used to increase the algorithm's resistance to plaintext attacks [9]. Therefore, hash functions such as MD5 and SHA-256 are widely used in the information security field. When any length of data is input, the output of a hash function is fixed in size. If the input of



(a) $x-y$, $x-w$ and $x-u$

(b) $x-y-w$, $x-y-z$ and $y-u-w$

Figure 2: The chaotic attractor in each plane of the 5D Lorenz hyperchaotic system

the hash function is modified slightly, the output will be completely different. Therefore, the attacker cannot infer the hash sequence to obtain plaintext image data [22]. Although the used hash function is pre-image resistance, in some encryption systems, the key is set very simply, and the attacker can successfully crack some messages through brute-force attacks [9]. To solve this problem, this paper combines MD5 and SHA-256 hash functions and adding a control parameter $KeyHex$ to encrypt the key multiple times to increases the security of the encryption algorithm.

Calculate the sum of the pixel values of all rows and columns of the plaintext image I with a size of $M \times N$, and record as $Sumrow$ and $Sumcol$ respectively. First, MD5 is used to calculate $Sumrow$, $SumCol$, and $KeyHex$, respectively. Then, the results are combined and calculated using SHA-256. Where $Keyhex$ = 6b679b3c71108d30a79e610526a8c18ef974c176f4e529f684748ac019931209. The method of calculating the mixed hash function is shown in Equation (3).

$$D = SHA-(256(MD5(SumRow)MD5(SumCol)MD5(KeyHex)). \quad (3)$$

2.3 DNA Coding and Operations

2.3.1 DNA coding

The four nucleic acid bases form the deoxyribonucleic acid (DNA) sequence [8]. The four binary numbers from 00 to 11 are used to represent A, C, G, and T, and only 8 of the 24 DNA coding schemes obtained to meet the conditions [21]. Table 1 shows the DNA coding rules. Table 2 shows examples of the application of DNA coding rules.

In the process of grayscale image encryption, the length of the binary sequence of each pixel is 8, and the corresponding DNA sequence length is 4. As shown in Table 2, the pixel grayscale value is 231 and its binary sequence is 11100111. If encoding and decoding according to different rules, completely different results will be obtained. In this paper, eight kinds of DNA coding rules are dynamically selected through the encryption key L_k. Coding operations are performed from top to bottom, left to right, and the random number obtained is 1 to 8. The calculation method is shown in Equation (4).

Table 1: DNA coding rules

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

$$E_{DC}^{(k)} = floor\,(8 * (L_k/3)) + 1 \quad (4)$$

where the function $floor(\cdot)$ indicates round down.

2.3.2 DNA Operations

DNA sequence operations are the same as binary operations [8, 21]. According to eight different DNA coding schemes, eight different DNA operation rules can be obtained. Each DNA coding rule corresponds to a DNA operation.

In this paper, the choice of DNA operation rules is determined by the encryption key L_k. The random selection rule of DNA operation is shown in Equation (5):

$$L_{operation} = floor\,(7 * (L_K/3)) + 1 \quad (5)$$

The corresponding relationship between the value of $L_{operation}$ and the operation rules is shown in Table 3.

Table 2: The examples of the application of DNA coding rules

Rules	1	2	3	4	5	6	7	8
231	TCGT	TGCT	GATG	GTAG	CATC	CTAC	AGCA	ACGA

Table 3: $L_{operation}$ and DNA operation rules comparison table

$L_{operation}$	1	2	3	4	5	6	7
operation rules	Addition	subtraction	multiplication	XNOR	XOR	left shift	right shift

The calculation results of Equation (5) are random numbers with values from 1 to 7, and corresponding operation rules can be carried out by referring to Table 3.

3 The Proposed Algorithm

Figure 3 shows the processing flow chart of the JPEG image encryption algorithm. The encryption algorithm is mainly realized through 5D Lorenze hyperchaotic system, mixed hash function (MD5 and SHA-256), DNA encoding/decoding and DNA operation.

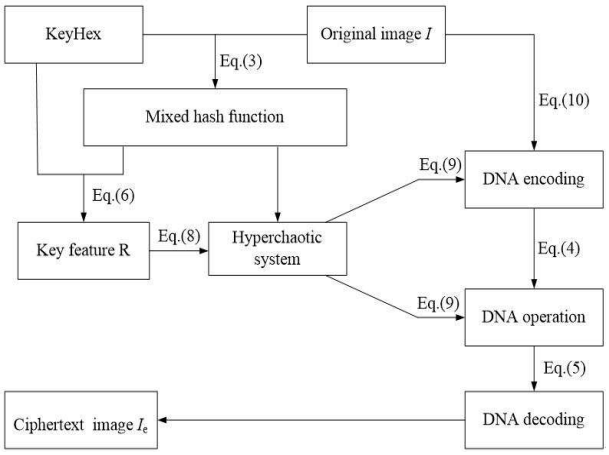


Figure 3: The processing flow chart of the JPEG image encryption algorithm

3.1 Generation of Key

In order to improve the resistance of image encryption algorithm to choice plaintext attack and known-plaintext attack, this paper adopts a mixed hash function to calculate the hash value of plaintext image as the initial parameter of a chaotic system. The specific steps of the key generation are as follows.

Step 1: Hash sequence D is calculated using the mixed hash function of Equation (3).

Step 2: Determine the value of R from the hash sequence D, and then determine the initial value of the hyperchaotic system by R. The calculation formula is shown in Equation (6).

$$R = hex2dec(D) \oplus hex2dec(KeyHex) \tag{6}$$

where $\oplus$ is XOR operation, $hex2dec(\cdot)$ reepresents the conversion from hexadecimal to decimal.

Step 3: Divide $hex2dec$ (D) into a group of 8 bits and get 32 segmented key d_i. Therefore, D' can also be expressed as:

$$D' = d_1, d_2, d_3, \cdots, d_{32} \tag{7}$$

Step 4: The initial value of chaos is obtained by Equation (8). The Equation (2) is continuously cycled $d_{31} + d_{32} + R$ times to obtain the chaotic sequence L with good randomness.

$$\mathrm{L} = \begin{cases} x_{(1)} = ((((d_3 \oplus d_4) \oplus (d_2 \oplus d_5)) \oplus (d_1 \oplus d_6)) \oplus R)/256 \\ y_{(1)} = ((((d_y \oplus d_{10}) \oplus (d_8 \oplus d_{11})) \oplus (d_7 \oplus d_{12})) \oplus R)/256 \\ z_{(1)} = ((((d_{15} \oplus d_{16}) \oplus (d_{14} \oplus d_{17})) \oplus (d_{13} \oplus d_{18})) \oplus R)/256 \\ u_{(1)} = ((((d_{21} \oplus d_{22}) \oplus (d_{20} \oplus d_{23})) \oplus (d_{19} \oplus d_{24})) \oplus R)/256 \\ w_{(1)} = ((((d_{27} \oplus d_{28}) \oplus (d_{26} \oplus d_{29})) \oplus (d_{25} \oplus d_{30})) \oplus R)/256 \end{cases} \tag{8}$$

Step 5: Chaotic sequence L can be calculated by Equation (9) to obtain the encryption key L_k.

$$L_k = \mod(floor(4 * L), 4) \tag{9}$$

The key generated by the above equation is closely related to the plaintext image. It can guarantee the unique key of each encryption process.

3.2 The Encryption Algorithm

Assume that the encrypted object is a grayscale image I of size $M \times N$. First, DNA encoding of the plaintext image which is controlled by the encryption key L_k. L_k is generated by a mixed hash algorithm (SHA-256 and MD5) and 5D Lorenz hyperchaotic system. Then, the encoded image is rearranged in the order of L_k to achieve pixel permutation. Finally, the encryption key L_k is used to dynamically select DNA operation rules, Perform relevant calculations, the results were decoded according to DNA coding rules to obtain the ciphertext image I_e.

The specific encryption processing steps are as follows:

Step 1: Obtain L_k through the key generation, and Two-dimensional matrix P is obtained from the original JPEG image.

Step 2: Divide P into 4 sub-blocks. The calculation method is shown in Equation (10).

$$E_{enc} = fix(P/4) \tag{10}$$

where the function $fix(\cdot)$ represents the rounding of the zero direction of the element in $P/4$.

Step 3: Encode each pixel value in E_{enc} using Equation (4) and the rules in Table 1.

Step 4: The L_k generated by the hyperchaotic system is sorted to obtain the sorted sequence L_p. The calculation method is shown in Equation (11):

$$(B, L_p) = sort(L_K) \tag{11}$$

where L_p is an array whose Size is equal to size(L_k). Each column of L_p is a permutation vector corresponding to the elements of column vectors in L_k.

Step 5: Transpose the subscript array L_p to get L'_p, and obtain the permutation image I_p through Equation (12).

$$I_p = image(L'_p) \tag{12}$$

where the function image($\cdot$) means to display L'_p as an image.

Step 6: Perform DNA operation on I_p and L_k according to the DNA operation rules in Equation (5) and Table 3.

Step 7: The DNA encoding and decoding rules in Equation (4) and Table 1 were used to decode the DNA operation results and obtain ciphertext image I_e.

3.3 The Decryption Algorithm

The receiver owns the ciphertext image I_e and the control parameter *KeyHex*. The decryption process of ciphertext image is as follows:

Step 1: Perform the key generation method in Section 3.1 to obtain R and L_k.

Step 2: Obtain the DNA coding sequence by executing Equation (10) and Equation (4).

Step 3: According to Equation (5) and the rules in Table 3, using the inverse operation to obtain the sequence L_p.

Step 4: Perform reverse permutation operation on L_p to get the decrypted image. Equation (13) is the calculation formula of permutation inverse operation.

$$I_{pi} = image(:) \tag{13}$$

4 Experimental Results and Analysis

The software simulation environment is MATLAB R2017a and the experimental platform is Intel (R)Core (TM) I5-2410m CPU @2.30ghz, 4.00GB RAM, Windows 7(64-bit) operating system. Four images Lena, Barbara, Baboon and Boat with a size of 512×512 were selected from USC-SIPI [20] image database as test images. To carry out the follow-up experimental work, OpenCV [15] is used to convert the image format to JPG. Figure 4 shows the original image, ciphertext image and decrypted image of test images.

4.1 Statistical Analysis

4.1.1 Histogram Analysis

The more even the distribution of the ciphertext image histogram, the higher the security of the encryption algorithm [5]. Figure 5 shows the histograms of plaintext images and ciphertext images.

As shown in Figure 5, the pixel values of the plaintext image are concentrated in some areas, while the pixel values in the ciphertext image histogram are uniformly distributed. This indicates that the statistical characteristics of digital images are destroyed and attackers cannot obtain any valid information related to plaintext through statistical attacks. Therefore, the proposed algorithm can resist the attack of statistical analysis.

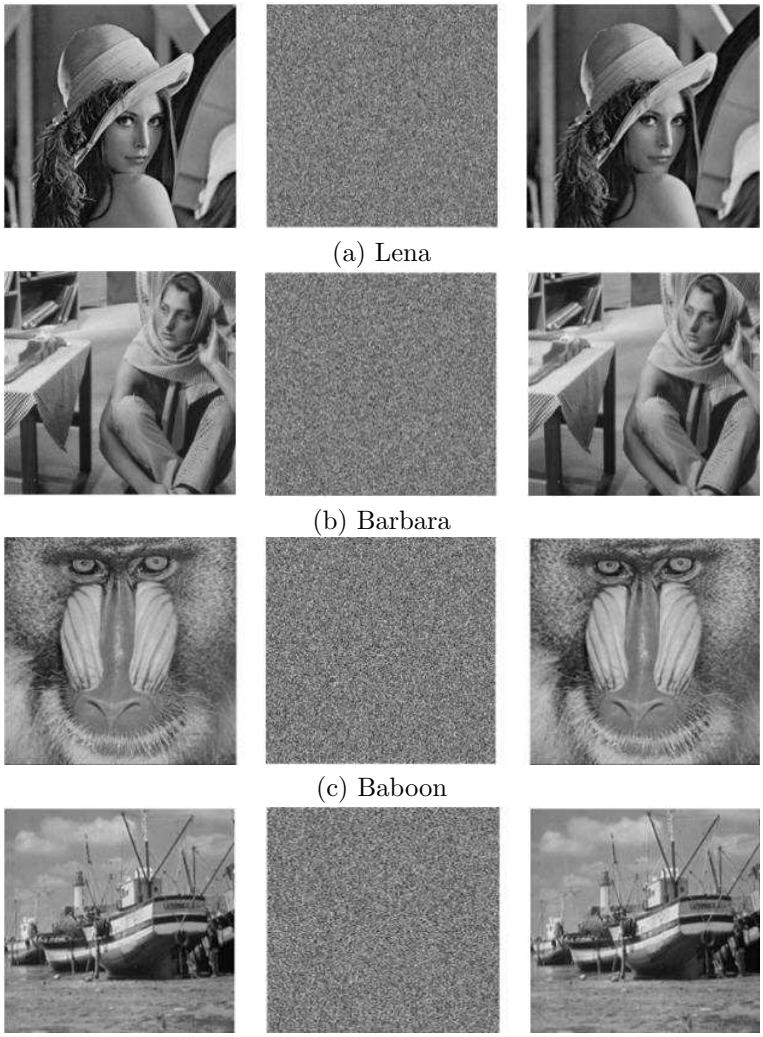
4.1.2 Correlation Analysis

It is necessary to weaken the strong correlation between adjacent pixels of original images to improve the security of multimedia data [26]. Three thousand pairs of pixels were randomly selected from three directions, and the correlation coefficient was calculated by the following equations:

$$\begin{aligned}
\rho_{xy} &= \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)} \times \sqrt{D(x)}} \\
\operatorname{cov}(x,y) &= \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)\left(y_i - E(y)\right) \\
E(x) &= \frac{1}{N}\sum_{i=1}^{N} x_i \\
D(x) &= \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2
\end{aligned}$$

where x and y respectively represent the gray value of two adjacent pixels, the total number is represented by N, the covariance of the variables x and y is represented by $\operatorname{cov}(x,y)$. The expectation and variance of the variable are represented by $E(\cdot)$ and $D(\cdot)$ respectively.

Table 4 shows the correlation between adjacent pixels of the plaintext image and ciphertext image. Table 5



(a) Lena

(b) Barbara

(c) Baboon

(d) Boat

Figure 4: The encryption/decryption results of test images

Table 4: Correlation between adjacent pixels of plaintext image and ciphertext image

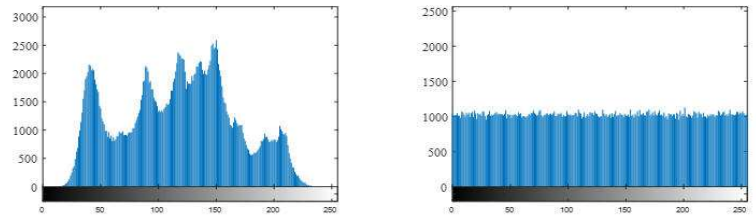
Test image	Plaintext image				Ciphertext image		
	Horizontal	Vertical	Diagonal		Horizontal	Vertical	Diagonal
Lena	0.9703	0.9841	0.9557		-0.0010	-0.0009	0.0007
Barbara	0.8899	0.9600	0.8859		0.0044	-0.0053	-0.0026
Baboon	0.8844	0.7633	0.7365		0.0079	-0.0001	-0.0009
Boat	0.9576	0.9801	0.9444		-0.0097	-0.0004	0.0060

shows the comparative analysis of the correlation between the proposed algorithm [6, 10, 11, 14]. Figure 6 shows the correlation between Lena plaintext image and its corresponding ciphertext image in all directions.

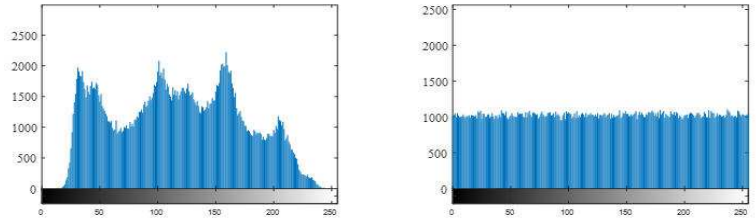
As shown in Table 4, the correlation coefficient of the plaintext image is close to 1, while the ciphertext image is closer to 0, and its value is much smaller than that of the plaintext image. As shown in Table 5, the resistance of the proposed algorithm to statistical attacks is significantly better than the encryption algorithm in [6, 10, 11, 14]. As shown in Figure 6, the pixels of the Lena plaintext image are clustered together, while the pixels of the ciphertext

Table 5: Correlation analysis between adjacent pixels of Lena ciphertext image

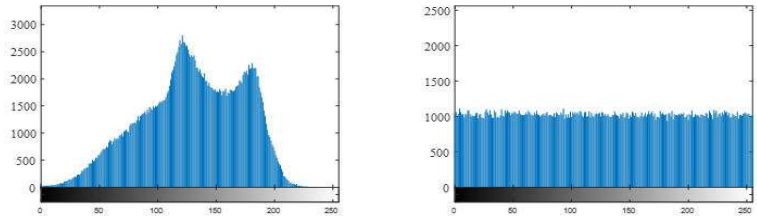
Methods		Horizontal	Vertical	Diagonal
[10]		-0.0025	0.0006	0.0072
[6]		-0.0033	-0.0040	-0.0002
[14]		0.0015	-0.0043	-0.0023
[11]		-0.3458	-0.0027	0.0033
Proposed		-0.0010	-0.0009	0.0007



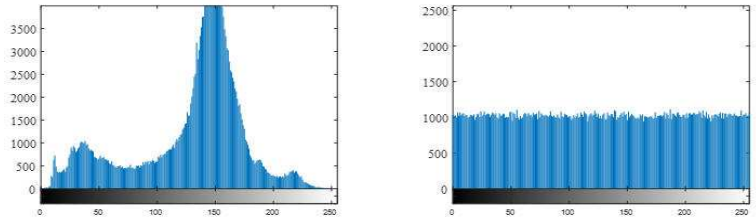
(a) Histogram of Lena



(b) Histogram of Barbara

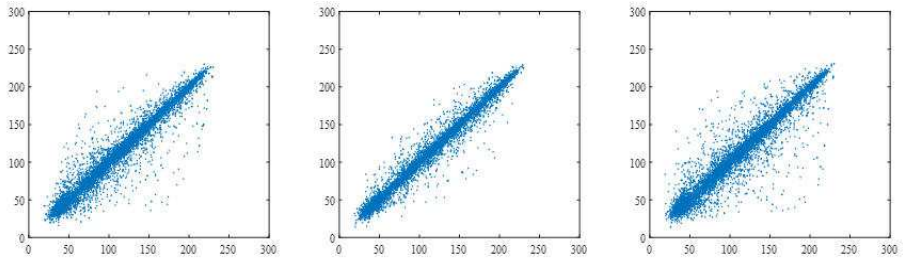


(c) Histogram of Baboon

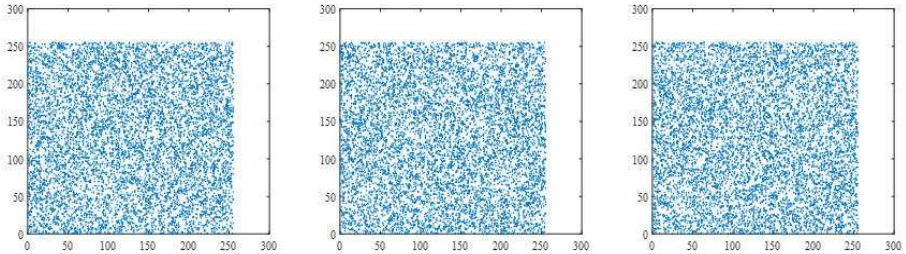


(d) Histogram of Boat

Figure 5: The histograms of plaintext images and ciphertext images of test images



(a) Horizontal, vertical, and diagonal directions of plaintext images



(b) Horizontal, vertical and diagonal directions of ciphertext images

Figure 6: Correlation comparison between plaintext image and ciphertext image of Lena image

Table 6: The entropy of the ciphertext image

	Lena	Barbara	Baboon	Boat
Original image	7.598324	7.632119	7.374090	7.203873
Ciphertext image	7.999314	7.999336	7.999262	7.999293

Table 7: Comparative analysis of information entropy of Lena ciphertext image

Methods	[7]	[6]	[14]	[11]	Proposed
Information entropy	7.80	7.9990	7.9993	7.8232	7.999314

Table 8: NPCR and UACI values of ciphertext images (100%)

Test image	Pixel position	Pixel value change	NPCR(%)	UACI(%)
Lena	(183,95)	$67 \rightarrow 68$	99.6117	33.4469
Barbara	(504,108)	$104 \rightarrow 105$	99.6159	33.4252
Baboon	(24,11)	$200 \rightarrow 201$	99.6315	33.4667
Boat	(368,180)	$35 \rightarrow 36$	99.6189	33.4474

image are more scattered. Therefore, the proposed algorithm can resist statistical analysis attacks.

4.2 Information Entropy Analysis

The information entropy [1] is calculated by Equation (14).

$$H = -\sum_{i=0}^{2^n-1} p_{(m_i)} \log_2 \frac{1}{p_{(m_i)}} \tag{14}$$

where the probability of occurrence of m_i is represented by $p(m_i)$.

Theoretically, the pixel values of the ciphertext image will have very good randomness when the information entropy is closer to 8. The information entropy of 4 test images is shown in Table 6. Table 7 compares and analyzes the entropy of Lena ciphertext image in the proposed algorithm [6,7,11,14].

As shown in Table 6, the information entropy of all test images is above 7.9992, close to the ideal entropy value of 8. Through the comparative analysis of information entropy in Table 7, it can be seen that the information entropy of the proposed algorithm is superior to the [6, 7, 11, 14]. Therefore, the proposed algorithm has good security to resist entropy attacks.

4.3 Differential Attack Analysis

The encryption algorithm is more resistant to differential attacks when the ciphertext image is more sensitive to the plaintext image, the number of pixels changes rate (NPCR) and the unified average changing intensity (UACI) [4,17] are often used as evaluation criteria. The calculation method is detailed in the following equations:

$$\begin{aligned} NPCR &= \frac{\sum_{i=0}^{M=1}\sum_{j=0}^{N=1} D(i,j)}{M \times N} \\ D_{(i,j)} &= \begin{cases} 0, c_1(i,j) = c_2(i,j) \\ 1, c_1(i,j) \neq c_2(i,j) \end{cases} \\ UACI &= \frac{1}{M \times N} \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \frac{|c_1(i,j) - c_2(i,j)|}{255} \end{aligned}$$

where the pixel values of the ciphertext image before and after changing the plaintext image are represented by $c_1(i,j)$ and $c_2(i,j)$ respectively, the length and width of the image are represented by M and N respectively.

Table 8 shows the NPCR and UACI values of the ciphertext image. Table 9 shows the comparison and analysis of NPCR and UACI in Lena ciphertext images of the proposed algorithm and [6,7,10,11,14].

Table 9: Different encryption algorithms NPCR and UACI comparison (100%)

Method	NPCR(%)	UACI(%)
[10]	95.43	21.34
[7]	99.45	27.03
[6]	99.57	33.35
[14]	99.6881	37.5600
[11]	99.6827	33.3781
Proposed	99.6117	33.4469

As shown in Table 8, the UACI values of different images are close to 33.46%, indicating that the pixel values have changed. The NPCR of different images is close to 100%, indicating that the position where the ciphertext image has changed. Therefore, the proposed algorithm can resistance differential attacks. As shown in Table 9, the proposed algorithm is compared with Lena ciphertext images of NPCR and UACI in the [6,7,10,11,14], the analysis shows that the proposed algorithm has better UACI and PSNR values, close to the optimal value, compared to [6,7,10,11,14].

4.4 Exhaustive Attack Analysis

4.4.1 Key Space Analysis

The more effective against brute-force attacks when the key space is larger [17]. Valid keys in the proposed algorithm are as follows:

1) Hash values generated by plaintext images using the mixed hash function;

2) Initial values of chaotic systems x_0, y_0, z_0, u_0 and w_0;

3) DNA coding rules (8 types).

When the computer precision is set to 10^{-15}, then the result of $2^{256}\times 8\times 10^{15}\times 10^{15}\times 10^{15}\times 10^{15}\times 10^{15} = 9.248\times 10^{152} \approx 2^{507}$ is the key space of the proposed algorithm. Table 10 shows the comparative analysis of the key space.

As shown in Table 10, compared with the key space in the [6, 10, 11, 14], the proposed algorithm is significantly larger. In contrast with the [7], the key space of the proposed algorithm is much smaller than that of the encryption scheme in [7]. This is because in the [7], each nonzero AC coefficient corresponds to an ACC, the number of ACCs in the entropy-coded data is large, When the number of ACCs involved in scrambling is much larger than the number of DCCs, this encryption step will result has a large key space. The key space of the proposed algorithm is approximately 2^{507} which is much larger than in 2^{100}. Therefore, the proposed algorithm can resist brute-force attacks.

4.4.2 Key Sensitivity Analysis

After slightly changing the encryption key, the corresponding result should be completely different [22]. Similarly, even minor changes to the encryption key cannot be able to get the correct plaintext image. For example, if the parameter x_0 of this algorithm is changed to $x_0 + 10^{-15}$, the correct decrypted image will not be obtained. The key sensitivity test analysis results of Lena ciphertext images are shown in Figure 7.

As can be seen from Figure 7, a slight modification to the key, the decryption results obtained are completely different. Moreover, if the attacker decrypts the image with an incorrect key, any valid feature information related to the original image will not be obtained.

4.5 Robustness Analysis

The anti-interference ability of an encryption system can be tested with robustness [16, 24, 25]. In the actual communication process, it is often affected by the transmission environment. To verify the robustness of the proposed algorithm, noise attacks, cropping attacks, and JPEG compression attacks will be used.

4.5.1 Noise Attack Analysis

In the image transmission process, inevitable noise will interfere with image decryption results [16, 24]. The robustness of the encryption algorithm is tested by adding noise of different intensities to the ciphertext image and using the same method for encryption and decryption operations. Figure 8 shows a Lena ciphertext image with different intensities of salt and pepper noise or Gaussian noise with a mean value of 0 and different variances and its corresponding decrypted image.

As can be seen from Figure 8, both can recover plaintext images, no matter how different kinds of noise are added. The image information is still identifiable, after adding salt and pepper noise with a noise intensity of 0.2 or adding Gaussian noise with a mean of 0 and a variance of 0.001. However, the image quality of the decrypted image after adding Gaussian noise is lower than that of adding salt and pepper noise.

4.5.2 Cropping Attack Analysis

Test the robustness of the algorithm against cropping attacks by cropping part of the encrypted image [25]. Fig.9 shows the ciphertext image and the decrypted image after cropping 1/16, 1/8, 1/4 and 1/2.

As can be seen from Figure 9. The decrypted image can still be identify after cropping the ciphertext image 1/2. Therefore, the proposed algorithm has better robustness for cropping attacks, and the image quality of decrypted images will be affected when the crop size is too large.

4.5.3 JPEG Compression

Perform JPEG compression on the ciphertext image to verify the robustness of the algorithm under JPEG compression. The measurement of JPEG compression adopts a quality factor, the range of quality factor is between 1 $\sim$ 100 [18]. Generally, the change of pixel value between plaintext and ciphertext image is measured by PSNR. The calculation method is shown as follows:

$$PSNR = 10 \times \log_{10}\left(\frac{M \times N \times 255^2}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(P(i,j) - C(i,j))^2}\right)$$

where the expected value in the plaintext image is represented by $P(i, j)$; the error value at the same position in the ciphertext image is represented by $C(i, j)$.

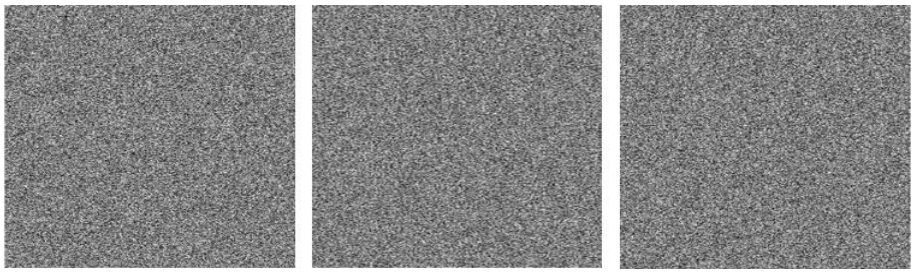
The calculation method for SSIM is defined as follows:

$$SSIM = \frac{\left(2u_p u_c + (0.01L)^2\right)\left(2\sigma_{pc} + (0.03L)^2\right)}{\left(u_p^2 + u_c^2 + (0.01L)^2\right) + \left(\sigma_p^2 + \sigma_c^2 + (0.03L)^2\right)}$$

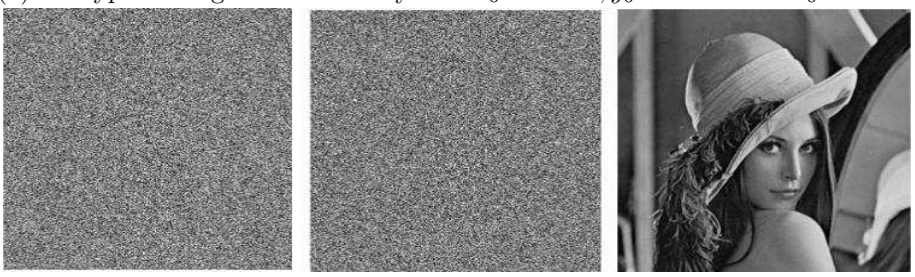
where the plaintext image and the restored image are represented by u_p and u_e respectively. The variance of the original image and the restored image is represented by δ_p and δ_c. δ_{pc} represents the covariance of the original and recovered images, and L represents the dynamic range of pixel values.

Table 10: Comparative analysis of key space

Method	[10]	[7]	[6]	[14]	[11]	Proposed
key space	$2^{256} \times 10^{42}$	3.15×10^{237206}	2^{224}	$2^2 56$	$2^2 56$	9.248×10^{152}

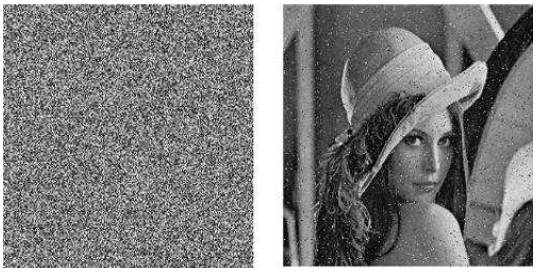


(a) Decrypted images when the keys are $x_0 + 10^{-15}, y_0 + 10^{-15}$ and $z_0 + 10^{-15}$

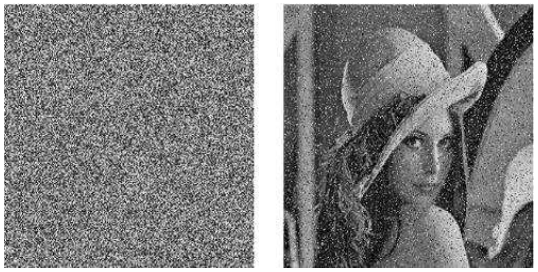


(b) Decrypted images when the keys are $u_0 + 10^{-15}, w_0 + 10^{-15}$ and correct key

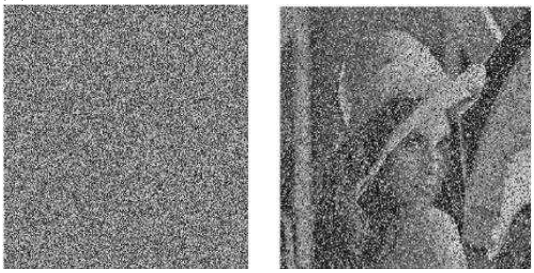
Figure 7: Key sensitivity analysis



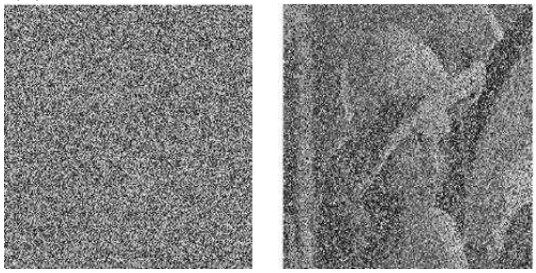
(a) The salt and pepper noise intensity of 0.05



(b) The salt and pepper noise intensity of 0.2

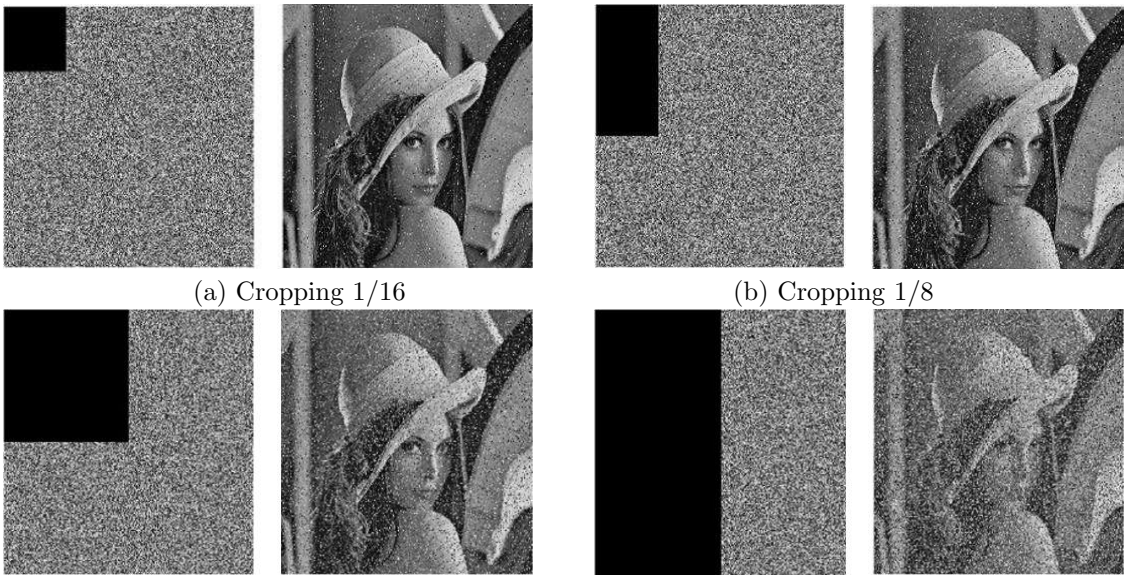


(c) The Gaussian noise variance 0.0001



(d) The Gaussian noise variance 0.001

Figure 8: Robustness analysis of Lena image encryption and decryption after adding Noise



(a) Cropping 1/16 (b) Cropping 1/8

(c) Cropping 1/4 (d) Cropping 1/2

Figure 9: Robustness analysis of Lena image encryption and decryption after cropping

Encrypts 4 images of Lena, Barbara, Baboon and Boat under different QF values with the proposed algorithm, and compared with the algorithm in Li's (2019) [10] and the algorithm in Li's (2017) [9]. Figure 10 shows the comparison result of average PSNR and SSIM of ciphertext image under different QF values.

As can be seen in Figure 10, the average PSNR values of the ciphertext image obtained by the proposed algorithm is maintained above 20 dB. Therefore, the security of encryption algorithm can be guaranteed. The average PSNR and SSIM values of [9] are lower than the proposed scheme and Algorithm 1 of [10]. This is because Algorithm 1 of [10] only performs encryption in the conversion phase and quantization phase. However, all the coefficients in [9] are encrypted, which sacrifices compression efficiency. Therefore, the proposed algorithm can resist JPEG compression attacks.

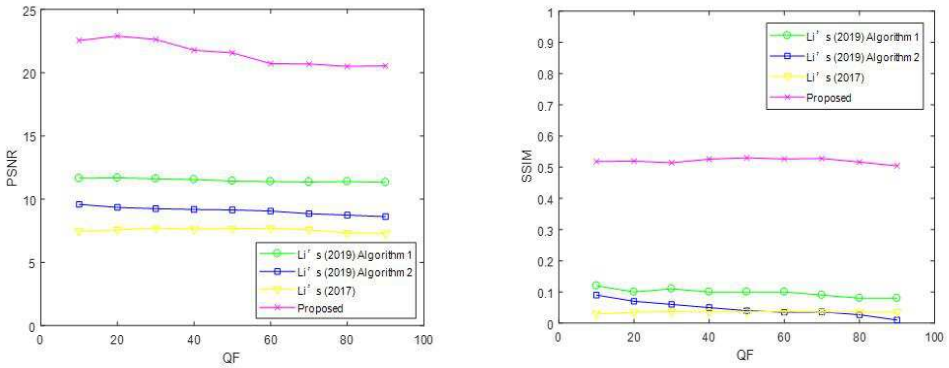
In order to further evaluate noise, cropping, JPEG compression attack after the decrypted image quality, it will have salt and pepper noise (0.05), Gaussian noise (0.0001), cropping (1/16), JPEG compression (QF = 50) after four kinds of attack the decrypted image PSNR and SSIM values and not subject to any attacks, comparing the decrypted image PSNR and SSIM values to evaluate the difference between different decrypted image, the results are shown in Table 11. Table 12 shows the comparison result of the average PSNR value of the decrypted image after the attack between the proposed algorithm and the method of [14].

As shown in Table 11, the decrypted image after being attacked by noise, cropping, and JPEG compression is compared with the decrypted image without any attack. It is found that the SSIM of the decrypted image before the attack is greater than 0.7300 and the PSNR remains above 40 dB. After adding noise, cropping and JPEG compression attacks, the values of SSIM and PSNR decrease significantly.

As can be seen in Table 12, the average PSNR values of the decrypted image after the noise attack of the proposed algorithm is close to or better than [14]. After the proposed algorithm suffers from cropping and JPEG compression attacks, the average PSNR values of the decrypted image is slightly weaker than the [14]. The results show that the decrypted image can still recover and identify when the ciphertext image is severely distorted. Therefore, the proposed algorithm has the ability to resist noise, cropping and JPEG compression attacks.

4.6 Time Complexity Analysis

The computational complexity of the proposed algorithm is mainly related to the encryption algorithm steps in Section 3.2. The time consumption of Step 1 is $O(4\times M\times N)$ because the time consumption is mainly the floating-point operation of generating chaotic sequences. Steps 2, 3, 4, and 5, the time complexity of each step is $O(4\times M\times N)$. This is because the time consumption is mainly the number of operations for image pixel conversion. The time complexity of Step 6 is $O(16\times M\times N)$. The time consumption is mainly the number of DNA operations. The time complexity of Step 7 is $O(16\times M\times N)$ because the time consumption is mainly the number of conversion operations of the DNA matrix. Therefore, the total time complexity of the proposed algorithm is $O(16\times M\times N)$. By testing the encryption algorithm of the proposed, the average encryption time of Lena image with a size of 512×512 is 2.573 s, which is better than the encryption time of 3.007 s in [14].



(a) Average PSNR values (b) Average SSIM values

Figure 10: Comparison of average PSNR and SSIM of ciphertext image under different QF values

Table 11: SSIM and PSNR values of ciphertext images before and after different attacks

decrypted image	Attack types	evaluation index	Lena	Barbara	Baboon	Boat
Before the attack	-	PSNR	48.501259	47.773492	46.993609	48.892712
		SSIM	0.8249	0.7403	0.7339	0.7323
After the attack	pepper and salt noise (0.05)	PSNR	27.943090	27.285805	26..670474	28.331530
		SSIM	0.5482	0.5280	0.6783	0.5264
	Gaussian noise (0.0001)	PSNR	26.946095	26.374115	26.046747	27.296853
		SSIM	0.5228	0.5075	0.5338	0.5040
	cropping (1/16)	PSNR	21.095905	20.075935	21.513170	21.406337
		SSIM	0.5398	0.5318	0.5881	0.5256
	JPEG compression (QF=50)	PSNR	22.249904	23.339515	20.403879	20.323898
		SSIM	0.5193	0.5042	0.5143	0.5301

Table 12: Comparison of average PSNR values of ciphertext images against noise, cropping and JPEG compression

Attack types	[14]	Proposed
Pepper and salt noise	26.5911	27.557725
Gaussian noise	27.2038	26.665953
Cropping	26.4020	21.022836
JPEG compression	27.9637	21.579299

5 Conclusions

In this paper, we have proposed a JPEG image encryption algorithm based on hyperchaotic, mixed hash and dynamic DNA, which improves the anti-attack ability and key space of the encryption algorithm and improves the efficiency and robustness of the existing encryption algorithm. Firstly, the encryption key is generated from the 5D hyperchaotic system, the mixed hash function and the original plaintext image. Secondly, the plaintext image is converted into a DNA matrix according to the DNA coding rules. The DNA encoded image is rearranged in order of the encryption key to achieving pixel permutation. Finally, the encryption key is used to dynamically select DNA operation rules. After relevant calculation, ciphertext images are obtained by decoding the results through DNA coding rules. The experimental results show that the proposed algorithm has large key space, high key sensitivity, high security and robustness, can resist noise, cropping , JPEG compression and other common attacks, and can be applied to secure image communication.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61862041, 61363078). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] A. Babaei, H. Motameni and R. Enayatifar, "A new permutation-diffusion-based image encryption technique using cellular automata and DNA sequence," *Optik*, vol. 203: 164000, 2020.

[2] P. Chaudhary, R. Gupta, A. Singh, P. Majumder and A. Pandey, "Joint image compression and encryption using a novel column-wise scanning and optimization algorithm," *Procedia Computer Science*, vol. 167, pp. 244–253, 2020.

[3] T. Chuman, K. Kurihara and H. Kiya, "On the security of block scrambling-based etc systems against jigsaw puzzle solver attacks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'17)*, pp. 2157–2161, June 2017.

[4] G. Z. Cui, L. F. Wang, X. C. Zhang and Z. Zhou, "An image encryption algorithm based on dynamic DNA coding and hyper-chaotic lorenz system," in *International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 226–238, 2018.

[5] R. Enayatifar, A. H. Abdullah, I. F. Isnin, A. Altameem and M. Lee, "Image encryption using a synchronous permutation-diffusion technique," *Optics and Lasers in Engineering*, vol. 90, pp. 146–154, 2017.

[6] M. Ghazvini, M. Mirzadi and N. Parvar, "A modified method for image encryption based on chaotic map and genetic algorithm," *Multimedia Tools and Applications*, vol. 79, no. 37, pp. 26927-26950, 2020.

[7] J. He, S. Huang, S. Tang and J. Huang, "JPEG image encryption with improved format compatibility and file size preservation," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2645–2658, 2018.

[8] M. Kar, A. Kumar, D. Nandi and M. K. Mandal, "Image encryption using DNA coding and hyper-chaotic system," *IETE Technical Review*, vol. 37, pp. 12–23, 2020.

[9] P. L. Li and K. T. Lo, "A content-adaptive joint image compression and encryption scheme," *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 1960–1972, 2017.

[10] P. L. Li and K. T. Lo, "Joint image encryption and compression schemes based on 16×16 DCT," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 12–24, 2019.

[11] Y. P. Li, C. H. Wang and H. Chen, "A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation," *Optics and Lasers in Engineering*, vol. 90, pp. 238–246, 2017.

[12] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[13] Z. Man, J. Li, X. Di and O. Bai, "An image segmentation encryption algorithm based on hybrid chaotic system," *IEEE Access*, vol. 7, pp. 103047–103058, 2019.

[14] B. Mondal, S. Singh and P. Kumar, "A secure image encryption scheme based on cellular automata and chaotic skew tent map," *Journal of Information Security and Applications*, vol. 45, pp. 117–130, 2019.

[15] Open Source Computer Vision Library[CP/OL]. (`https://opencv.org/`)

[16] A. U. Rehman and X. F. Liao, "A novel robust dual diffusion/confusion encryption technique for color image based on chaos, DNA and SHA-2," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 2105-2133, 2019.

[17] M. R. Salman and A. K. Farhan, "Color image encryption depend on DNA operation and chaotic system," in *First International Conference of Computer and Applied Sciences (CAS'19)*, pp. 267–272, 2019.

[18] W. Sirichotedumrong, T. Chuman, S. Imaizumi and H. Kiya, "Grayscale-based block scrambling image encryption for social networking services," in *IEEE International Conference on Multimedia and Expo (ICME'18)*, pp. 1-6, 2018.

[19] V. Thanikaiselvan, S. Patel and S. Sivanantham, "Secured data transmission through dual domain reversible data hiding and encryption in images," in *International Conference on Inventive Computation Technologies (ICICT'20)*, pp. 840–847, Feb. 2020.

[20] USC-SIPI Image Database[DB/OL]. (`http://sipi.usc.edu/database/`)

[21] X. Y. Wang, Y. Wang, X. Q. Zhu and S. Unar, "Image encryption scheme based on chaos and DNA plane operation," *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 26111–26128, 2019.

[22] X. Y. Wang, X. Q. Zhu, X. J. Wu and Y. Q. Zhang, "Image encryption algorithm based on multiple mixed hash functions and cyclic shift," *Optics and Lasers in Engineering*, vol. 107, pp. 370–379, 2018.

[23] Z. R. Wang, W. Li, B. L. Zhu and X. Q. Li, "A joint image lossless compression and encryption method based on chaotic map," *Multimedia Tools and Applications*, vol. 76, no. 12, pp. 13995–14020, 2017.

[24] G. D. Ye and X. L. Huang, "A secure image encryption algorithm based on chaotic maps and SHA-3," *Security and Communication Networks*, vol. 9, no. 13, pp. 2015-2023, 2016.

[25] Q. Y. Zhang, J. T. Han and Y. T. Ye, "Image encryption algorithm based on image hashing, improved chaotic mapping and DNA coding," *IET Image Processing*, vol. 13, no. 14, 2019.

[26] X. C. Zhang, Z. Zhou, Y. Niu, Y. F. Wang and L. F. Wang, "An image encryption algorithm based on chaotic system using DNA sequence operations," in *International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 213–225, 2018.

Biography

Zhang Qiu-yu. Researcher/Ph.D. supervisor, graduated from Gansu University of Technology in 1986, and then worked at school of computer and communication in Lanzhou University of Technology. He is vice dean of Gansu manufacturing information engineering research center, a CCF senior member, a member of IEEE and ACM. His research interests include network and information security, information hiding and steganalysis, multimedia communication technology.

Ye Yu-tong. received the BS degrees in communication engineering from Shanghai Normal University, Shanghai, China, in 2016. Her research interests include network

and information security, information hiding and steganalysis.